1992

Improved Methodology for Limit States Finite Element Analysis of Lattice Type Structures using Nonlinear Post-Buckling Member Performance

Markus Ostendorp
Portland State University

10.15760/etd.1178

IMPROVED METHODOLOGY FOR LIMIT STATES FINITE ELEMENT ANALYSIS OF LATTICE TYPE STRUCTURES USING NONLINEAR POST-BUCKLING MEMBER PERFORMANCE

by

MARKUS OSTENDORP

A dissertation submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY
in
SYSTEMS SCIENCE\CIVIL ENGINEERING

Portland State University
1992

TO THE OFFICE OF GRADUATE STUDIES:

The members of the Committee approve the dissertation of Markus Ostendorp presented June 23, 1992.

[redacted]

Wendelin H. Mueller, Chair

[redacted]

Franz N. Rad

[redacted]

Trevor D. Smith

[redacted]

George G. Lendaris

[redacted]

Hormoz Zarefar

APPROVED:

[redacted]

Roy Koch, Director, Systems Science Ph.D. Program

[redacted]

C. William Savery, Interim Vice Provost for Graduate Studies and Research

AN ABSTRACT OF THE DISSERTATION OF Markus Ostendorp for the Doctor of Philosophy in Systems Science\Civil Engineering presented June 23, 1992.

Title: Improved Methodology for Limit States Finite Element Analysis of Lattice Type Structures using Nonlinear Post-Buckling Member Performance.

APPROVED BY THE MEMBERS OF THE DISSERTATION COMMITTEE:

[redacted]

Wendelin H. Mueller, Chair

[redacted]

Franz N. Rad

[redacted]

Trevor D. Smith

[redacted]

George G. Lendaris

[redacted]

Hormoz Zarefar

In an attempt to achieve more efficient designs, the technological frontier is pushed further and further. Every year science probes for a better understanding of natural phenomena, discovering new and improved methods to perform the same task more efficiently and with better results. One

of the new technologies is the nonlinear analysis of structural systems using inelastic post-buckling member performance. Inelastic post-buckling member performance is defined as the constitutive relationship between axial load and displacement after the ultimate member capacity has been exceeded. A nonlinear analysis is able to predict the failure behavior of a structural system under ultimate loads more accurately than the traditionally used linear elastic analysis. Consequently, designs can be improved and become more efficient, which reduces the realization cost of a project.

An improved nonlinear analysis solution algorithm has been developed, that allows the analyst to perform a nonlinear analysis using post-buckling member performances faster than previously possible. Furthermore, the original post-buckling member performance database was expanded using results obtained from physical member compression tests. Based on the experimental results, new post-buckling member performance model curves were developed to be used together with the improved nonlinear solution algorithm.

In addition, a program was developed that allows the analyst to perform a valid nonlinear analysis using a finite element program (LIMIT). The program combines a numerical pre-processor, and input and output data evaluation modules based on human expertise together with the LIMIT analysis package. Extensive on-line help facilities together with graphical pre- and post-processors were also integrated into the program. The resulting analysis package essentially combines all of the necessary components required to perform a nonlinear analysis using post-buckling member performances into one complete analysis package.

ACKNOWLEDGEMENTS

This investigation was carried out under the supervision of Dr. Wendelin H. Mueller. I am indebted to Dr. Mueller for providing me relentlessly with guidance, support, and valuable advice throughout the duration of the research project.

Special thanks and appreciation goes to the Civil Engineering Department at Portland State University and in particular to the Department Head, Dr. Franz Rad, and Dr. Trevor Smith, whose advice I always value.

In addition, I would like to thank Dr. Hormoz Zarefar, Mechanical Engineering Department, and Dr. George Lendaris, Systems Science Ph.D. program core faculty for their contribution to the success of this research project.

Furthermore, I would like to show my gratitude to Leon Kempner Jr., a senior engineer at Bonneville Power Administration, one of the sponsors of the research project.

TABLE OF CONTENTS

PAGE

ACKNOWLEDGEMENTS.................................... iii

LIST OF TABLES...................................... vii

LIST OF FIGURES.....................................viii

CHAPTER

I INTRODUCTION.................................. 1

Theory of Nonlinear Analysis........... 1

Conventional Approaches to Nonlinear Finite Element Analysis........... 7

LIMIT - A Nonlinear Finite Element Analysis Program.................. 10

CURVEPLOT - Post-Buckling Member Performance Database.............. 16

Example of a Limit States Analysis..... 19

Expert System Methodology and Concepts...................... 22

Focus of Research...................... 31

Literature Review...................... 33

II LIMITATIONS OF CURRENT PRACTICE............. 50

Degree of Complexity................... 50

Modeling Errors........................ 60

Hidden and Numerical Algorithm Related Errors.................... 64

Interpretation of Results.............. 66

Human Factors and Resources............ 67

III RESEARCH METHODOLOGY AND PROCESS............ 69

Solution to Current Limitations........ 69

Development of the
Limit States Analysis Module...... 74

Changing the Role of NEXPERT OBJECT.... 82

Development of the Limit States
Analysis Module Components........ 89

Modification of CURVEPLOT.............. 93

Expansion of the Post-Buckling Member
Performance Database.............. 95

Development of the Towerplot
Graphics Module................... 102

Development of the Loadhistory
Graphics Module................... 104

Development of the New Limit States
Analysis Algorithm................ 105

Modification of the LIMIT
Analysis Program.................. 109

Development of the
Bandwidth Optimization Module..... 112

Revision of the LIMIT User Manual...... 114

Development of the
LIMIT Modeling Manual............. 115

IV DEVELOPMENT OF THE EVALUATION COMPONENTS
OF THE LIMIT STATES ANALYSIS MODULE......... 130

Extraction of the Knowledge............ 130

Description of the Knowledge........... 133

Structuring and Encoding the Knowledge. 151

Interaction of the Evaluations and the
LIMIT Nonlinear Analysis Program.. 160

Verification and Validation of the
Limit States Analysis Module...... 163

Implementation of the Limit States Analysis Module in Design and Analysis.................... 167

V SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS... 170

Summary.............................. 170

Conclusions.......................... 176

Recommendations...................... 179

REFERENCES.. 183

GLOSSARY OF TERMS.................................... 186

APPENDICES

A EXISTING EXPERT SYSTEM APPLICATIONS......... 194

B SELECTION OF EXPERT SYSTEM DEVELOPMENT SOFTWARE....................... 198

C DESCRIPTION OF THE LIMIT STATES ANALYSIS MODULE................ 219

D FILE AND DIRECTORY STRUCTURE................ 254

LIST OF TABLES

TABLE PAGE

I Summary of Review of Expert System Software Packages...................... 218

LIST OF FIGURES

FIGURE		PAGE
1.	Generalized 2-Force Member..................	35
2.	Constitutive Law............................	36
3.	Linear Constitutive Law.....................	37
4.	Nonlinear Constitutive Law..................	38
5.	Loading and Unloading Path..................	39
6.	Equivalent Strain Energies U_1 and U_2........	40
7.	Tangent Stiffness Method....................	41
8.	Secant Stiffness Method.....................	42
9.	Strain Energy Violation of Secant Method....	43
10.	Experimental Test Results - KL/R=60.........	44
11.	Experimental Test Results - KL/R=120........	45
12.	Experimental Test Results - KL/R=210........	46
13.	Normalized Bilinear Member Performance......	47
14.	Normalized Member Performance Curves........	48
15.	Simplified Structure........................	49
16.	Functional Description of the Limit States Analysis Module Components.............	117
17.	Recommended Normalized Curves 1-10..........	118
18.	Recommended Normalized Curves 11-15.........	119
19.	Recommended Normalized Curves 16-20.........	120
20.	Recommended Normalized Curves 21-25.........	121
21.	Recommended Normalized Curves 26-30.........	122

22. Constant Force Member Performance........... 123

23. Experimental Test Results - KL/R=60......... 124

24. Experimental Test Results - KL/R=120........ 125

25. Experimental Test Results - KL/R=210........ 126

26. Regression Data Base - Equal Leg Angles..... 127

27. Regression Data Base - Unequal Leg Angles... 128

28. Regression Data Base - Double Angles........ 129

29. Limit States Analysis Module Credit Screen.. 235

30. Limit States Analysis Module Disclaimer..... 236

31. Limit States Analysis Module Main Menu...... 237

32. Limit States Analysis Module
Input Directory........................... 238

33. Limit States Analysis Module
Output Directory.......................... 239

34. Limit States Analysis Module
View\Print File........................... 240

35. Limit States Analysis Module
Limit File Viewer......................... 241

36. Limit States Analysis Module Input Menu..... 242

37. Limit States Analysis Module
Title Line Editor......................... 243

38. Limit States Analysis Module
Control Parameters........................ 244

39. Limit States Analysis Module Member Editor.. 245

40. Limit States Analysis Module
Dead Load Editor.......................... 246

41. Limit States Analysis Module

Joint Load Editor...................... 247

42. Limit States Analysis Module

Specified Deflection Editor............ 248

43. Limit States Analysis Module

Bandwidth Optimization................. 249

44. Limit States Analysis Module Load History... 250

45. Limit States Analysis Module

Towerplot\Curveplot Opening 251

46. Limit States Analysis Module Towerplot...... 252

47. Limit States Analysis Module Curveplot...... 253

CHAPTER I

INTRODUCTION

THEORY OF NONLINEAR ANALYSIS

Within the field of structural analysis, there exist a variety of models which attempt to represent and predict the real life behavior of structural components and their interaction within the structural system. In order to explain and discuss the theory of nonlinear analysis some of the fundamental concepts involved with respect to the relatively simple idealized two-force member will first be presented.

A two-force member is a structural model, which assumes that the behavior of a load carrying component can be completely specified by the location (i.e. coordinates in space) of its starting and ending joint, and the three translational degrees of freedom associated with each of the joints (See Figure 1). In addition information is needed about the magnitudes and directions of the forces applied at the joints, and about an either experimentally obtained, or mathematically derived constitutive law relating strains to stresses and therefore displacements to forces. An example of a constitutive law is shown in Figure 2. The translational degrees of freedom allow for displacements of the initial joint position parallel to the three orthogonal directions which define the observation space. The two-force member model assumes that forces are only applied at the joints, and that the force remains constant throughout the length of the member. In addition, the model assumes that the resulting member force is either tensile (i.e. attempting to pull the member apart), or compressive (i.e. attempting to shorten the member), and that it will act through the centroid of the geometric cross-section in the long direction of the member (i.e. load is not applied eccentrically). Furthermore, it is assumed that the stress (compressive or tensile) is distributed uniformly throughout any cross-section along the length of the member.

The constitutive law is a mathematical function, usually determined experimentally, which relates strains (i.e. the ratio of absolute deformation to original length) with stresses (i.e. force per unit area). This relationship between strains and stresses may be a linear function (i.e. proportional) as shown in Figure 3, or a nonlinear (i.e. not proportional) function as shown in Figure 4. At the same time, the constitutive law may be influenced by the material characteristics, and may possess different mathematical relationships between stress and strain for the loading and unloading processes (See Figure 5). If the relationship between stress and strain is the same for both, the loading and unloading process, the material is said to behave elastically. If the relationship between stress and strain is not the same, the material is said to perform inelastically. Consequently, three fundamental constitutive concepts can be identified, and will be presented in the following paragraphs.

A linear elastic structural analysis, also known as a 1 st order analysis, assumes that the constitutive relationship between stress and strain is linear, and that the slope of the function for any strain level is a numerical constant. This numerical constant is defined as the differential change in stress with respect to a differential change in strain. It is named "Young's Modulus of Elasticity (E)" in honor of its discoverer and can be determined experimentally for most well behaved construction materials. In addition, since it is assumed that the material behaves elastically, there is no need to specify if the force is applied gradually to, or gradually taken off the member, since the relationship that exists between stress and strain is unique. The concept of linear elasticity provides an efficient means to perform a quick, relatively easy to perform, and reasonably accurate analysis of the behavior of a structure. However, this concept will not account for the reserve capacity of a structural system because it does not take into account the energy that is dissipated in the members due to their permanent deformations. It has been determined experimentally that for most construction materials the assumption of linear elasticity is only valid if the applied loads are small enough so that the resulting deformations do not induce a permanently set strain in the material (i.e. a strain that makes the material yield). Even though most structural analyses assume linear elastic behavior, it has been determined, that the results obtained will be inaccurate if the stress level exceeds the linear portion of the stress-strain relationship. In particular, the forces calculated in the members will be higher than their respective ultimate capacities, and the calculated deflections will be smaller than the true deflections. This difference is shown in Figure 6, in which the strain energy of the model U_1 should equal U_2, the strain energy of the real stress-strain relationship (i.e. the areas under the two curves should have an equal magnitude).

In addition, it has been determined that the strength and behavior of a structure is really dependent upon the ultimate or collapse load. The ultimate or collapse load of a structural system is defined as the load at which failure of the system becomes imminent (i.e. the structural system is at the "Limit State"). At the limit state, the capacity of the structure equals the load demands imposed upon the system and any further increase in the load demands will cause the structure to become unstable and to collapse. Therefore, the "Factor of Safety" (F.S.), which is defined as the ratio of the structural capacity and the imposed load demands, will equal unity at the limit state. On the contrary, in a traditional elastic analysis based on allowable stress design, the F.S. is selected to equal less than unity. Consequently, the computed structural capacity (via the linear elastic analysis) is reduced to a fraction of the true capacity in an attempt to stay well below the limit state of the system and to avoid failure.

In order to obtain a better model for the real material behavior, it is assumed in a nonlinear elastic analysis (also known as a 1 st order inelastic analysis), that the mathematical relationship between stress and strain is not linear, and that the modulus of elasticity (i.e. the slope of the mathematical relationship) varies as a function of the strain. As a result, it is not usually possible to determine the member load directly for a certain strain level, since both, the strain and the corresponding modulus are necessary to calculate the appropriate member load. Iterative numerical search methods usually have to be utilized to determine the correct combination of strains, modulus values, and stresses, which will satisfy the boundary conditions of the structural system. These iterative search methods require a large number of computations, which are mostly performed by computer systems. Analyses, based on the nonlinear elastic model, are more accurate, and predict a performance more comparable to the true behavior and strength of the structural system. This type of analysis is better able to predict the collapse load factor in a full scale tower load test than the elastic analysis method. Furthermore, since the analyst is able to account for the reserve capacity of the material, designs can be made less conservative, and more cost effective.

In a 1 st order inelastic analysis, the modulus value is a function of the strain, and it is different for the loading process than it is for the unloading process. The member load is therefore a function of the strain level, the modulus value, and the previous stress level. This phenomena can also be modelled as a member load versus displacement response. Again, iterative numerical search methods have to be employed to determine a solution, which will require an even greater number of computations than is necessary in the nonlinear elastic method.

CONVENTIONAL APPROACHES TO NONLINEAR FINITE ELEMENT ANALYSIS

With the advent of the computer technology it is possible to perform computation intensive nonlinear analyses for even large structural systems in a relatively short amount of time. A variety of iterative search techniques have been developed throughout the years to obtain solutions to problems using nonlinear constitutive relationships. The most common of these iterative search techniques, including their advantages and disadvantages are presented in the following paragraphs.

In the tangent method, the constitutive relationship is assumed to be piece-wise linear (4, 5, 15, 16, 17, 18, 19). The iterative solution algorithm increases the stress and strain levels in increments, and computes the corresponding modulus values as shown in Figure 7. Member deflections and forces are calculated for each of the increments, and added to the results obtained from the previous incremental computation. Stability and compatibility checks are performed following each incremental computation. Pending the results of the equilibrium checks, the iterative algorithm determines the final solution, or proceeds to perform the next incremental computation. The major advantage of the tangent method results from the fact that any nonlinear relationship can be divided into piece-wise linear segments if the size of the step is small enough. Nevertheless, the tangent method can not handle constitutive relations with negative modulus values (i.e. negative slopes), making it an unsuitable method to model buckling effects of compression members (i.e. the loss of capacity after the ultimate capacity of the member has been exceeded).

The secant method, on the other hand, does not calculate stiffnesses in increments, but rather calculates the stiffness as a function of a unique strain level (4, 5, 15, 16, 17, 18, 19). The trial stiffness is calculated to be the slope of the line, which intercepts the origin of the stress-strain relationship, and the point on the curve corresponding to the selected strain level. This is illustrated in Figure 8. The iterative search mechanism calculates the successive trial stiffnesses, deflections, and forces, until it finds the set of values which satisfies the specified force and deflection boundary conditions. The major advantage of the secant method results from the fact that it is able to deal with the negative slopes computed from the constitutive relationship within its solution algorithm since the stiffnesses that are computed are always positive. The disadvantage accrues from the fact that the secant method violates (to some extent) fundamental energy principles, since it does not approximate the path of the stress-strain relationship as shown in Figure 9. However, experiments based on full scale tower load tests (4, 24, 25) have shown that the secant method produces a reasonably close approximation to the true solution.

The LIMIT (1) finite element program utilizes a modified version of the secant method in its solution algorithm which is called the "direct stiffness method". It is called the direct stiffness method because it uses a member performance curve (i.e. uses a force-displacement relationship) rather than a constitutive relationship (i.e. a stress-strain relationship). The LIMIT program is the old VAX\VMS batch environment version which will be referred to as LIMIT B in the document. During the course of the research it was necessary to make major changes to the LIMIT B program to adapt it to the expert system. The changes made to the program have been documented in the new LIMIT (22) user manual. The modified version of the LIMIT program integrated into the Limit States Analysis Module will be referred to in this document as LIMIT ES.

LIMIT - A NONLINEAR FINITE ELEMENT ANALYSIS PROGRAM

LIMIT B is a three dimensional truss analysis program (written in the FORTRAN 77 programming language), which is able to consider the effects of the nonlinear post-buckling behavior of two force members in its analysis. Two force members are structural elements used to model members, which are assumed to be loaded in either compression or tension rather than in bending. Members stressed in compression or tension behave nonlinear if deformations become significant. LIMIT B is able to utilize the user specified nonlinear post-buckling behavior of the members (i.e. by using experimentally derived post-buckling member performance curves) to arrive at a solution for the forces and resulting deflections in a structure. Examples for experimentally derived post-buckling member performance curves for a short, intermediate, and long member are shown in Figures 10, 11, and 12 respectively.

The LIMIT B program can be used to perform both, an elastic and nonlinear analysis. In an elastic analysis, the program will directly proceed to establish the stiffness parameters, connectivity, and boundary conditions. The program will then solve for the resulting joint displacements in the structure and compute the final member forces. Since member capacities are a part of the LIMIT B program input, LIMIT B will also flag all of the overstressed members. In a nonlinear analysis, LIMIT B will calculate the joint displacements and member forces based on an elastic member behavior. A numerical iteration algorithm is utilized to check and update stiffness parameters until convergence (within user specified tolerance) on a particular solution which satisfies all of the specified boundary conditions is achieved.

There are three different types of nonlinear analyses that can be performed using the LIMIT B program. The three types are:

- Bilinear Performance Analysis: The post-buckling member performance is assumed to behave linear elastic, perfectly plastic for both compression and tension as shown in Figure 13. LIMIT B will perform the analysis using the previously mentioned iterative solution algorithm and compute a collapse load factor for the structural system. The output produced by LIMIT B in a nonlinear analysis contains information about members behaving elastic or inelastic, displacements, and member forces.

- Normalized Performance Analysis: The post-buckling member performance used as input for the LIMIT B program in a normalized analysis consists of a set of 11 different model performance curves as shown in Figure 14. Depending upon the member characteristics normalized model curves are assigned by the user to all the members that will behave inelastically during the nonlinear analysis. These model curves are converted to actual performances using the properties of the members.

- Actual Performance Analysis: The post-buckling performance used by the LIMIT B program in the nonlinear analysis is based upon experimental data derived from physical structural member tests (5). Actual member performances are defined in terms of real load and deflection values rather than normalized values. Typical actual member performance curves for a short, intermediate, and long member are shown in Figures 10, 11, and 12 respectively.

Typically, a nonlinear analysis using the LIMIT B program would proceed in the manner described in the following paragraphs.

First, the input data file format and the structural model are checked by performing an elastic analysis with the LIMIT B program and comparing the results calculated with the results calculated by another structural analysis tool such as the TOWER (2) program. This first analysis also allows the engineer to identify the members that are most critically stressed, since a part of the input to LIMIT B contains the member's tension and compression capacity information.

Next, the user would add specific control parameters that are necessary to run a nonlinear analysis with LIMIT B using bilinear member performance curves. In addition, the user has to flag any of the applied loads that are supposed to be increased during the nonlinear analysis. Only the loads that are flagged will be increased during the nonlinear analysis (i.e. the collapse load factor calculated by LIMIT B is the sum of the increased loads multiplied by the collapse load factor plus the sum of all the constant applied loads). A bilinear member performance curve number is then assigned to the most critically stressed member and the analysis is run.

Typically, the analysis process never stopped despite the fact that many of the members that had a bilinear member performance curve assigned to it showed large deformations. The process of identifying the critical member, assigning a bilinear member performance curve, and running the nonlinear LIMIT B analysis using bilinear member performance curves continues until LIMIT B is able to calculate a collapse load factor. At this point, the user checks the individual member performances, the control parameters are tightened and the whole process described above repeats until LIMIT B is again able to calculate a collapse load factor. The collapse load factor that is calculated is different for each set of control parameters.

Eventually, many analyses later, the results calculated by LIMIT B will not change anymore indicating that a final solution for the collapse load factor of the structural model assuming bilinear member behavior has been found. In order to further refine the nonlinear analysis the user has to run now a nonlinear LIMIT B analysis using normalized nonlinear member performance curves as described in the following paragraphs.

In order to run the first nonlinear analysis using LIMIT B and normalized member performance curves the user has to add certain control parameters and assign normalized member performance model curves to any of the members that have been identified as being critical during the bilinear analysis. As mentioned previously, the user has a choice of eleven nonlinear normalized member performance model curves. The user makes a guess, based on member strength and geometric characteristics, on which of the normalized model curves of the set of eleven to assign to the member. The user then compares the guess against an experimentally derived member performance curve contained in the CURVEPLOT (3) member performance database which will be discussed in more detail in the following section. If the selection of the normalized model curve seems adequate, the user can proceed to guess a curve for the next critical member, or else guess again until the most appropriate normalized model curve is found. Once every member has been assigned a normalized model curve number the LIMIT B analysis using the normalized performance curves is run. The results are then reviewed to determine if the selection of the normalized curves was appropriate for the points at which these members are performing at (i.e. does the normalized nonlinear model curve match the nonlinear post-buckling member performance curve predicted by the CURVEPLOT database at the load and displacement level calculated by LIMIT B). If the selection does not seem appropriate a different model curve is assigned and the analysis is rerun. Typically, it takes many iterations until an appropriate set of normalized performance curves is selected and a valid collapse load factor is calculated. Many iterations are usually required since the choice of the control parameters greatly influences the selection of the appropriate set of nonlinear curves and vice versa.

CURVEPLOT - POST-BUCKLING MEMBER PERFORMANCE DATABASE

CURVEPLOT is a graphical pre-processor program (written in FORTRAN 77 programming language) for the old Limit States Analysis Program (LIMIT B). The program contains an experimentally derived data base which characterizes the nonlinear behavior of structural components used in lattice structures such as transmission towers. The database contains member performance information obtained from experimental tests (5) of steel angle members in compression with slenderness ratios varying from 200 to 350 (i.e. it contains information about long, slender compression members only).

In addition, the program contains information about the set of eleven normalized nonlinear behavior models (piece-wise linear performance curves), which can be selected individually by the user based on the true behavior characteristics of the experimentally tested structural members. These normalized nonlinear member performance models are required as input for a nonlinear analysis using the LIMIT B program.

A typical session of CURVEPLOT entails the start-up of the program, followed by the display of the main menu. The user is prompted to answer questions regarding the strength and geometric properties of the structural member under study, and to make a selection of one of the eleven normalized nonlinear member performance models. The program will then proceed to display the experimentally derived post-buckling member performance curve together with the selected normalized nonlinear behavior model. The user is then able to compare the post-buckling member performance curve predicted by the CURVEPLOT database with the selected normalized nonlinear model curve. If the two curves appear similar in shape and scale, the user may choose to select the displayed model curve or can exit the program to try a different nonlinear model curve.

Typically, it takes the user a few times of running the CURVEPLOT program until an acceptable normalized model curve is selected for each individual member. The process of selecting nonlinear model curves continues until a curve has been selected for each individual member.

Once the analysis using the normalized nonlinear model curves has been run, the user has to verify the selection of each of the individual curves as mentioned in the previous section. The user has to perform a lengthy series of calculations and graphical manipulations to check each individual nonlinear model curve selected to determine if it is adequately close to the post-buckling member performance curve predicted by the CURVEPLOT database. If the two curves coincide well, the selection was appropriate. If the curves do not match well, a different model curve has to be selected and the analysis has to be rerun. Typically, it will take a few iterations using the LIMIT B program until a final and valid solution for the collapse load factor is obtained.

Many aspects are included in the selection of the appropriate nonlinear behavior model, requiring the user to be experienced with the nonlinear finite element analysis method and the way CURVEPLOT works. A lack of experience by the user with these subjects can easily result in a faulty analysis of the structural system under study; and it is this issue which will be discussed in detail in subsequent sections. One of the main purposes of the research presented in this document is to alleviate these problems in such a manner that an inexperienced engineer can perform a nonlinear analysis and select appropriate member performance curves.

The LIMIT B was originally developed to be used as a research tool. The purpose of the LIMIT B program was to predict the collapse load factors of transmission towers tested in full scale load tests. In its original version, the LIMIT B program was very cumbersome and could only be used by experts. Much experience with the subject and the program was necessary to use the program in a proficient manner and to select the appropriate performance curves.

EXAMPLE OF A LIMIT STATES ANALYSIS

In this section, an example is presented to explain the nonlinear limit states analysis that LIMIT B performs to obtain a solution for the collapse load factor of a structural system.

Consider the simplified structure shown in Figure 15(a). A force P is applied horizontally at the upper right corner. Assume that members 1-2, 2-3, 3-4, and 1-4 are very large and strong enough so that they are not significantly stressed for any reasonable load P. The applied load P is resisted by a tensile force T in member 1-3 and a compressive force C in member 2-4. Member 1-3 has a nonlinear tensile member performance as shown in Figure 15(c). Furthermore, member 2-4 has a nonlinear compressive member performance as shown in Figure 15(b).

Note, that both member performances are shown as a relationship of axial load versus displacement and it can be assumed that they have been obtained through experimental member tests and have been approximated by piece-wise linear line segments (i.e. the true experimental curves have been converted to nonlinear model curves just like post-buckling member performance curves from CURVEPLOT would be modelled by one of the eleven nonlinear model curves shown in Figure 14). The member performance curves therefore reflect the influence of all variables such as nonlinear material behavior, eccentric axial loading, intermediate joints, initial crookedness, and other, before and after failure of the member occurs.

From Figure 15(b) one can determine that the ultimate compression capacity of member 2-4 equals 5 kips, and that the ultimate tension capacity of member 1-3 equals 15 kips. If an elastic analysis is performed it can be determined that C does not equal T (i.e. the magnitude of the force in member 1-3 will not be equal to the magnitude of the force in member 2-4). This phenomena can be attributed to the difference in the slopes of the tension and compression member performance models. Since the ultimate compressive capacity of member 2-4 equals 5 kips, the applied load P_1 can be calculated to equal:

$$P_1 = (5+4)*\mathrm{Cos}(45^{o}) = 6.36 \text{ kips} \qquad (1)$$

for an axial member displacement X_1. P_1 is defined as the elastic capacity of the structure because any further increment of the load P will induce a permanently set strain in the compression member.

If one assumes that P_1 is further increased to cause an axial member displacement X_2 one can then calculate the allowable load P_2 to be equal to:

$$P_2 = (5+5.3)*\text{Cos}(45^o) = 7.28 \text{ kips} \qquad (2)$$

which is larger than the previously calculated elastic capacity of the structure. Even though the capacity of member 2-4 remained constant it can be seen that the total applied load capacity of the structural system increased, since member 1-3 had not reached its capacity.

This phenomena constitutes the basic underlying principle of a limit states analysis using the LIMIT B program. In addition, one can continue the analysis by calculating structural system capacities P_i for subsequent member displacements X_i as follows:

$$P_3 = (2+12.9)*\text{Cos}(45^o) = 10.53 \text{ kips} \qquad (3)$$

$$P_4 = (1.7+15)*\text{Cos}(45^o) = 11.81 \text{ kips} \qquad (4)$$

$$P_5 = (1.0+15)*\text{Cos}(45^o) = 11.31 \text{ kips} \qquad (5)$$

$$P_6 = (0.0+15)*\text{Cos}(45^o) = 10.60 \text{ kips.} \qquad (6)$$

A plot of structural capacities P versus member displacements X is shown in Figure 15(d). One can determine from the plot that the maximum capacity of the structure is P_4 = 11.81 kips at an axial member displacement X_4. Therefore, one can conclude that the limit states structural capacity P_4 is about 1.9 times higher than the elastic capacity P_1 of the structure. In addition, one can determine from Figure 15(d) that any further increase in the applied load P at a displacement X_4 can not be supported by the structure (i.e. the structure is at its limit state) since it would make the structure become unstable and cause collapse.

EXPERT SYSTEM METHODOLOGY AND CONCEPTS

Artificial intelligence is the science of modeling the human reasoning process. This science attempts to provide ways for automating processes which would usually otherwise require human intelligence and expertise. Artificial intelligence utilizes symbolic and logic reasoning rather than numbers and definitions in its approach of modeling the human thinking process.

There are a variety of important application areas within the artificial intelligence field, such as robotics, computer vision, speech synthesis, automated reasoning and theorem proving, natural language processing, neural networks, and expert systems. Out of all of these areas in the artificial intelligence field, expert systems is the one which has produced the most practical applications.

Most of today's more complicated problems can not be solved solely through the application of some "cookbook solution algorithms", but will in addition require some experience with the subject under study. In many cases, there exists a need for the analyst to use previously acquired experience and some measure of intuition in order to solve a problem successfully.

Expert systems are computer programs which attempt to simulate this intuitive portion of the human problem solving process. Expert systems (included in the more general designation of knowledge based systems) are software programs usually written in a fourth generation computer language that use rules of thumb to solve complex problems through logic reasoning rather than the application of numerical algorithms. Typically, these programs are highly user friendly as a result of the simple-to-understand fourth generation programming language, and/or the extensive use of graphical interfaces. Similar to human experts, expert systems provide advice to the user by calling upon available knowledge, or by asking for supplemental information needed to solve a problem intelligently and appropriately.

The "knowledge base" the expert system utilizes to arrive at a solution is usually stored as a series of IF/THEN type rules in combination with objects, classes, and properties of the objects, specifically developed for a particular problem context. Supplemental information is supplied by the user in response to queries raised by the expert system. The "intelligence" of the expert system is derived from the use of efficient mechanisms which process and evaluate the information given by the user and the data and knowledge base in order to arrive at a solution or conclusion for a particular problem. This mechanism is called an "inference engine". So it may be stated, that there are two basic components which can be identified in almost all rule based expert systems:

1) The Knowledge Base
2) The Inference Engine.

The knowledge base is the collection of the known information in the form of rules. Each rule includes one or more conditional statements, and may include multiple conclusions. If the conditions of the statements are met, the expert system will assume the associated conclusion(s) as true and store them for subsequent use.

The inference engine is a computer program that examines the knowledge base and processes queries or responses of the user. In the inference engine, rules are "fired" in accordance with a set order established by the developer of the expert system. Inference engines can be classified as backward chaining or forward chaining. Backward chaining inference engines, which are goal driven, arrive at conclusions by evaluating what supporting conditions must be true to arrive at the specified goal. Forward chaining inference engines, which are rule driven, utilize some known initial conditions in order to determine the final solutions possible with the specified given information.

Expert system technology should be considered for all those situations where it is important to preserve expertise and experience, or where the human experts need some relief from repetitive decision making tasks. Expert systems can be used in many instances, such as:

- To provide expertise if it is scarce, expensive, and not immediately or sufficiently available.

- To improve the work performance of unskilled personnel, and to increase the efficiency and productivity of skilled personnel.

- To define and preserve expert knowledge for situations where human expertise may be lost due to retirement, reassignment, economic reasons, illnesses, and vacations.

- To minimize the amount of errors produced by the handling of large numbers of repetitive tasks, thereby improving the overall quality and consistency of the results.

- To guarantee that expertise is applied uniformly, objectively, and consistently at all times, resulting in a more uniform company policy and/or regulations.

Therefore, one can conclude that the expert systems inference mechanism in combination with the expert knowledge base will improve the performance and quality, increase productivity, promote uniformity of results and policy, preserve essential knowledge for times to come, and utilize human resources more efficiently.

A variety of other aspects have to be discussed in order to present a complete picture of the expert system methodology and concepts. Some of the more important aspects are:

- An expert is any person who, through training and intimate experience with a certain subject matter, has achieved a degree of skill in dealing with the subject matter that is beneficial to capture and preserve.

- The skill being captured within the expert system does not necessarily have to be something complex, but rather may deal with simple but time consuming every day tasks.

- It is very unlikely that an expert system will ever be able to completely match the performance of a human expert in dealing with complex tasks. However, the expert system will aid the human expert by increasing the expert's productivity, by improving the consistency and completeness of the results obtained, and by minimizing repetitive task related errors.

- Expert systems are able to outperform an individual human expert, if the expert system combines the knowledge and experience of multiple experts. This will be particularly beneficial if the expertise combined within the expert system is extracted from complimentary sub-domains within one area.

An expert system can be programmed in such a manner that the developer will be able to specify a variety of different roles or controls the expert system can assume. For example, the expert system can be instructed to assume any of the following roles:

- The expert system can analyze a situation, make all necessary decisions, and act independently without requiring any human intervention. In this case, it would act as a completely autonomous expert.

- The expert system can analyze a situation, propose all necessary decisions to the user, and accept either a user override or a user acceptance of those decisions. In this case, it would be acting as an autonomous expert with human override.

- The expert system can act as an expert consultant to the user, suggesting related information, proposing additional ideas, providing recommendations on unusual situations, and suggesting possible causes of actions. In this case, the program could be characterized as an expert consultant.

- The expert system can act as an intelligent assistant to the user, making suggestions, pursuing alternatives, advising of consequences, and freeing the user of repetitive tasks.

- The expert system can act as a work horse for the user, performing repetitive and time consuming tasks, thereby freeing the user to work on the more complex parts of the situation or problem.

The development of an expert system entails four fundamental steps. These steps can be grouped in the following manner:

- The selection of an appropriate application area (Domain Selection), i.e. an area which requires one or more of the characteristics of the expert system programming advantages, which can not be dealt with using conventional programming.

- The selection of one or more human experts within the area of focus (Selection of Experts).

- The determination of all relevant techniques, knowledge, and heuristics usually used by the human expert(s) to solve the task being considered for the expert system (Knowledge Acquisition).

- The design and implementation of a computer program which is representative of the process the human expert employs to deal with the selected application area (Program Development).

Some of the other related areas with expert system development are the selection of the software tool, selection of the hardware platform, selection of the method chosen for representation of the knowledge, selection of the way that the knowledge will be implemented, and selection of the method chosen for the implementation and testing of the developed program. All of these areas and the terminology used in the expert system field will be further discussed in subsequent sections. As an aid to the reader, artificial intelligence related terminology used herein is defined alphabetically in the "Glossary of Terms", following the Reference Section.

FOCUS OF RESEARCH

The purpose of the research presented in this document was to develop an improved nonlinear analysis solution algorithm that would facilitate the creation of an expert system application that would combine all of the aspects of the nonlinear analysis methodology with the required tools. The primary goals of the improved nonlinear solution algorithm that was developed was to reduce the time necessary to perform an analysis, to reduce the level of complexity of the analysis, and to produce more consistent results.

A test program was developed to expand the original post-buckling member performance database. As previously mentioned, the post-buckling member performances are used as input to the LIMIT B program. However, the improved nonlinear analysis solution algorithm required the original member performance database to be expanded. Compression test results of short, single angles with equal and unequal legs and short, intermediate, and long double angles were used to expand the post-buckling member performance database. Based on the test results, a new set of thirty post-buckling member performance model curves was developed to facilitate the use of the improved nonlinear analysis solution algorithm.

In addition, the improved nonlinear solution algorithm was to be integrated into the development of a complete nonlinear analysis package. The purpose of this development was to enable an inexperienced and also an experienced user to perform an analysis using the improved solution algorithm. An inexperienced user of the LIMIT program is defined as an individual which understands the principles of structural analysis but had no prior exposure to nonlinear analysis concepts or the LIMIT program.

An expert system application was developed and has been named the Limit States Analysis Module. The expertise contained in the program has been solicited from human experts. The expertise was utilized within a variety of program components, namely the control program, the pre-processor, the graphic modules, and the input and output evaluation modules.

Using the improved solution algorithm the Limit States Analysis Module essentially acts as an expert consultant to the user. In this role, the Limit States Analysis Module shows related information, proposes additional ideas, and suggests possible courses of action. In addition, the Limit States Analysis Module checks the validity of the model and the accuracy of the results calculated by the program.

LITERATURE REVIEW

The literature review presented focused on expert system applications as a way to determine if any improved nonlinear analysis methodologies had been developed. A number of expert system applications have been recently, and are currently being developed in the area of structural analysis and structural mechanics (6, 7, 8, 9 , 10, 11, 12, 13, 14). The majority of these expert system applications that have been and are currently being developed address only one aspect of the structural analysis process.

Expert system technology is currently used to maintain control of numerical solution algorithms during runtime executions of finite element analysis programs (6, 7, 9, 10); to assist the user of analysis programs in the preparation of the required data input files for specific structural analysis programs (8, 11, 12, 13); to advise the user on how to create an optimum finite element mesh for a specific analysis problem (8, 11, 12); to assist the user in the process of selecting the best solution strategy for a specific structural analysis problem (6, 11); to provide assistance to the user during the model development phase and model validation process for a specific analysis problem (11, 12); and\or to help the user in the selection of the most appropriate finite element code to be used for a specific structural analysis problem (6). A more detailed description of existing expert system applications in the field of structural analysis that are characteristic of current development trends is provided in Appendix A.

Most of the expert systems that have been and are currently being developed in the field of structural analysis assume that the material behaves linear elastic. Only one expert system application (7) could be located in the literature that assisted a user during a structural analysis using nonlinear material behavior. This particular expert system application assists the user interactively in maintaining control over the nonlinear finite element solution algorithm used in the structural analysis program.

Currently, there do not exist (i.e. no mention of an existing application could be located in the literature) any expert system applications in the field of structural analysis that assist an engineer throughout all phases of a nonlinear structural analysis. Furthermore, the author was not able to locate any documented expert system applications that assist the user in the nonlinear analysis process using experimentally derived post-buckling member performances.

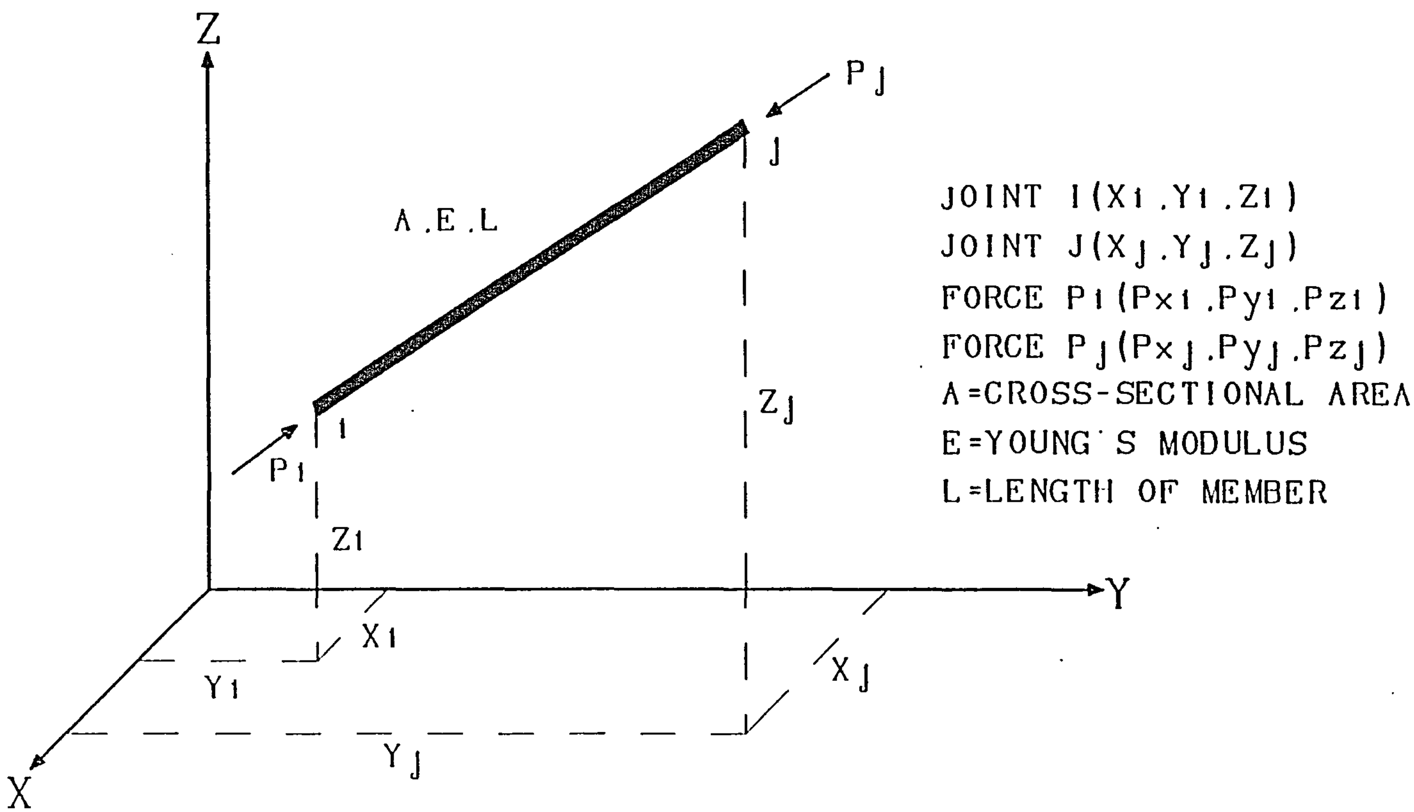


Figure 1. Generalized 2-Force Member.

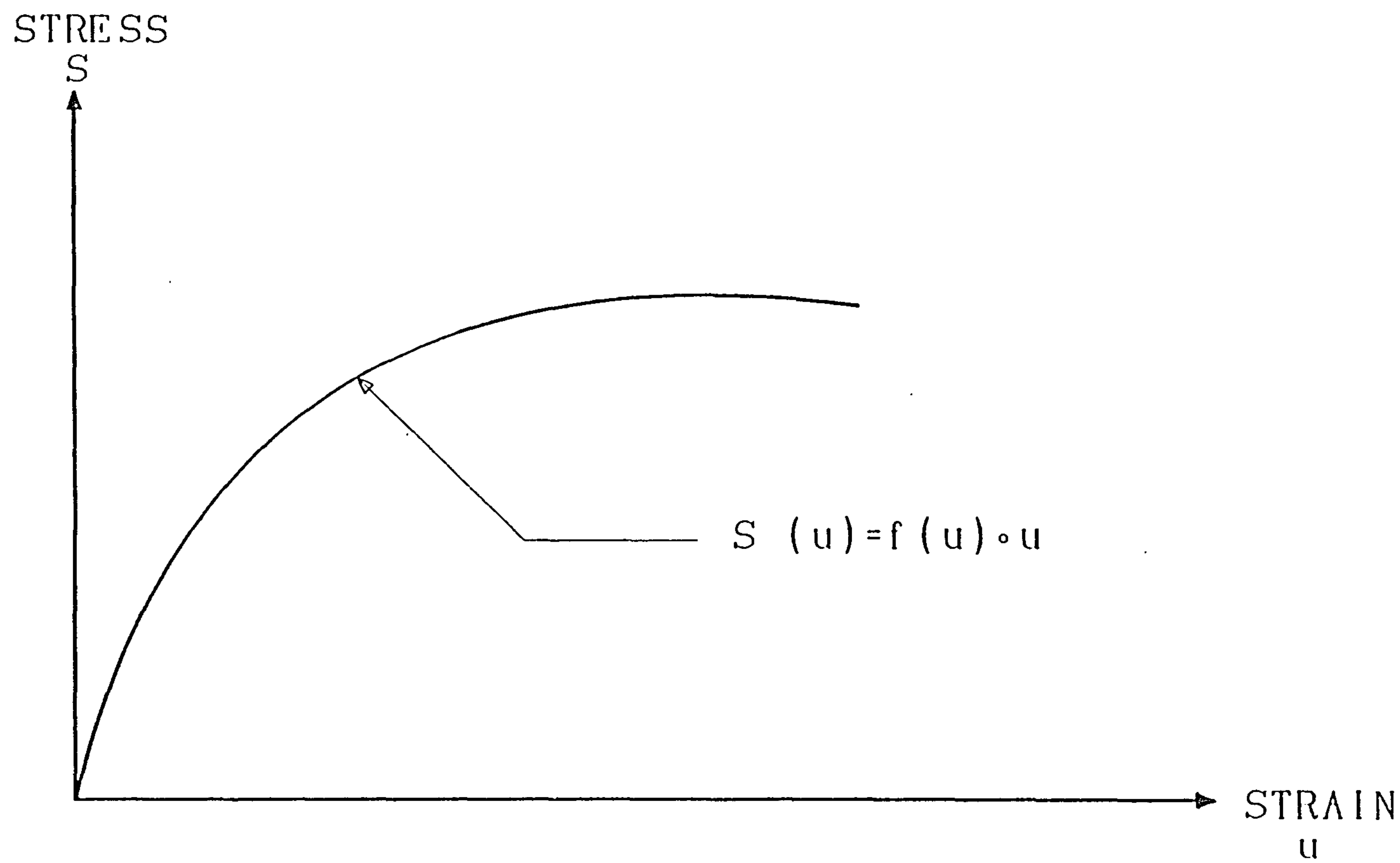


Figure 2. Constitutive Law.

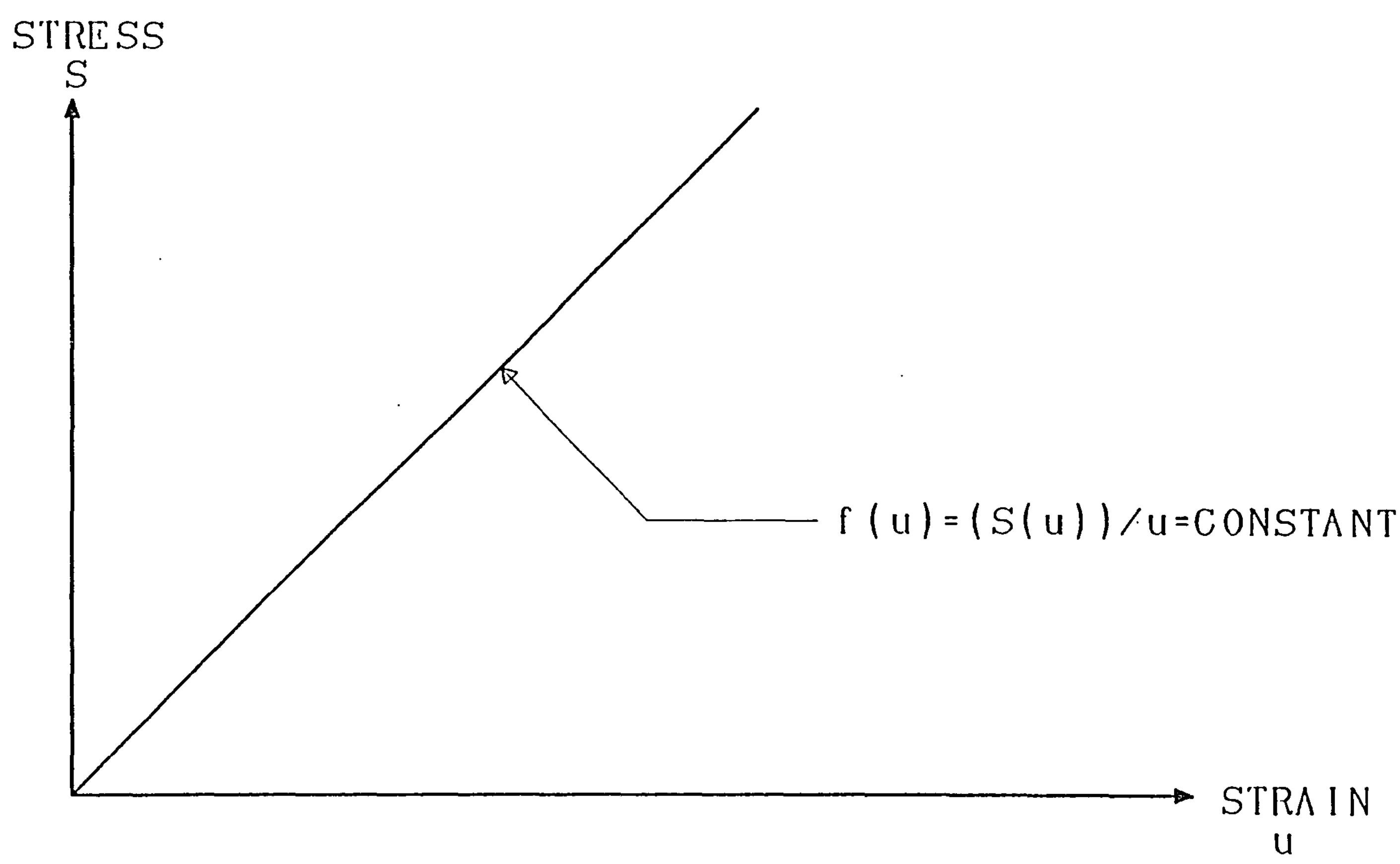


Figure 3. Linear Constitutive Law.

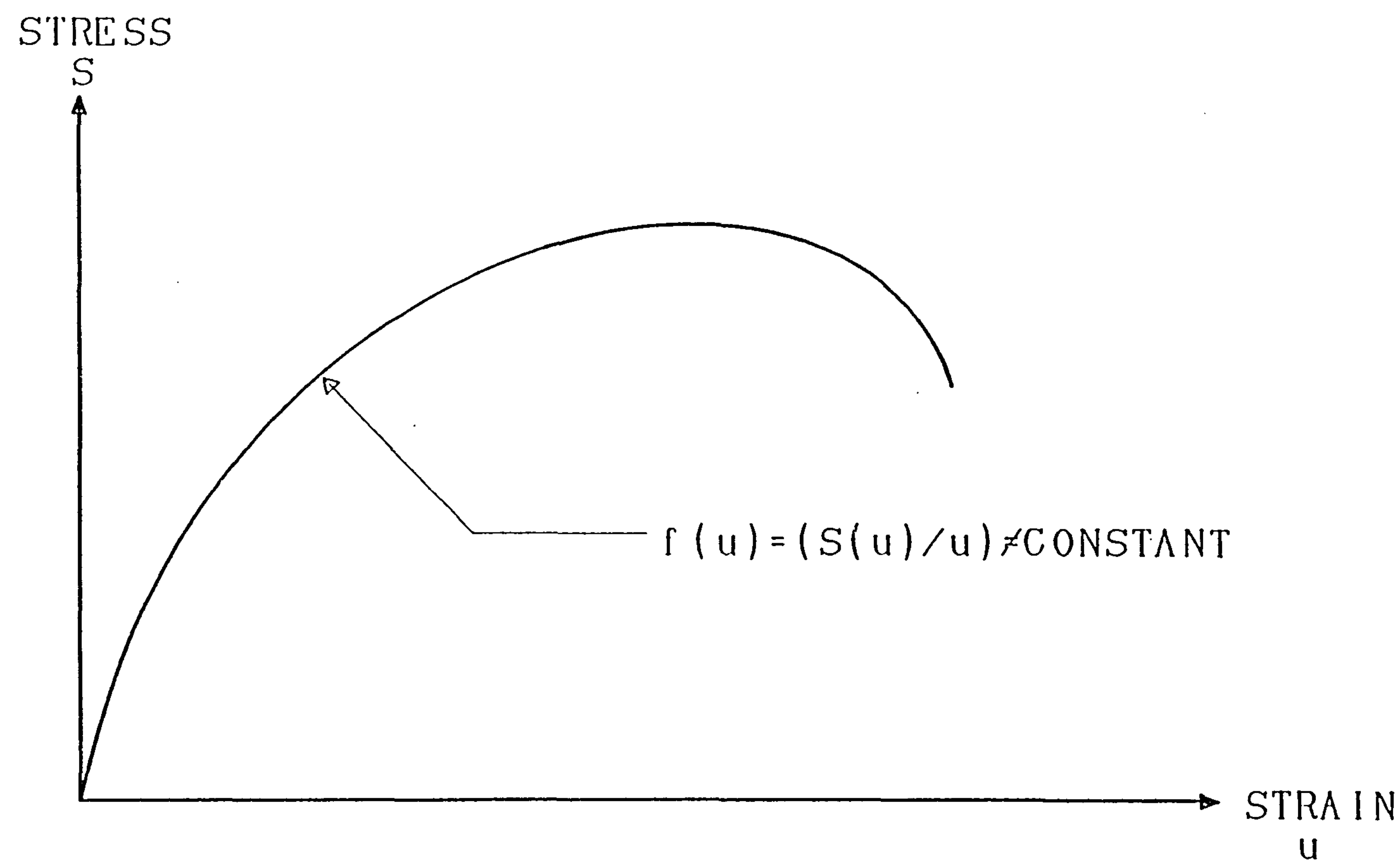


Figure 4. Nonlinear Constitutive Law.

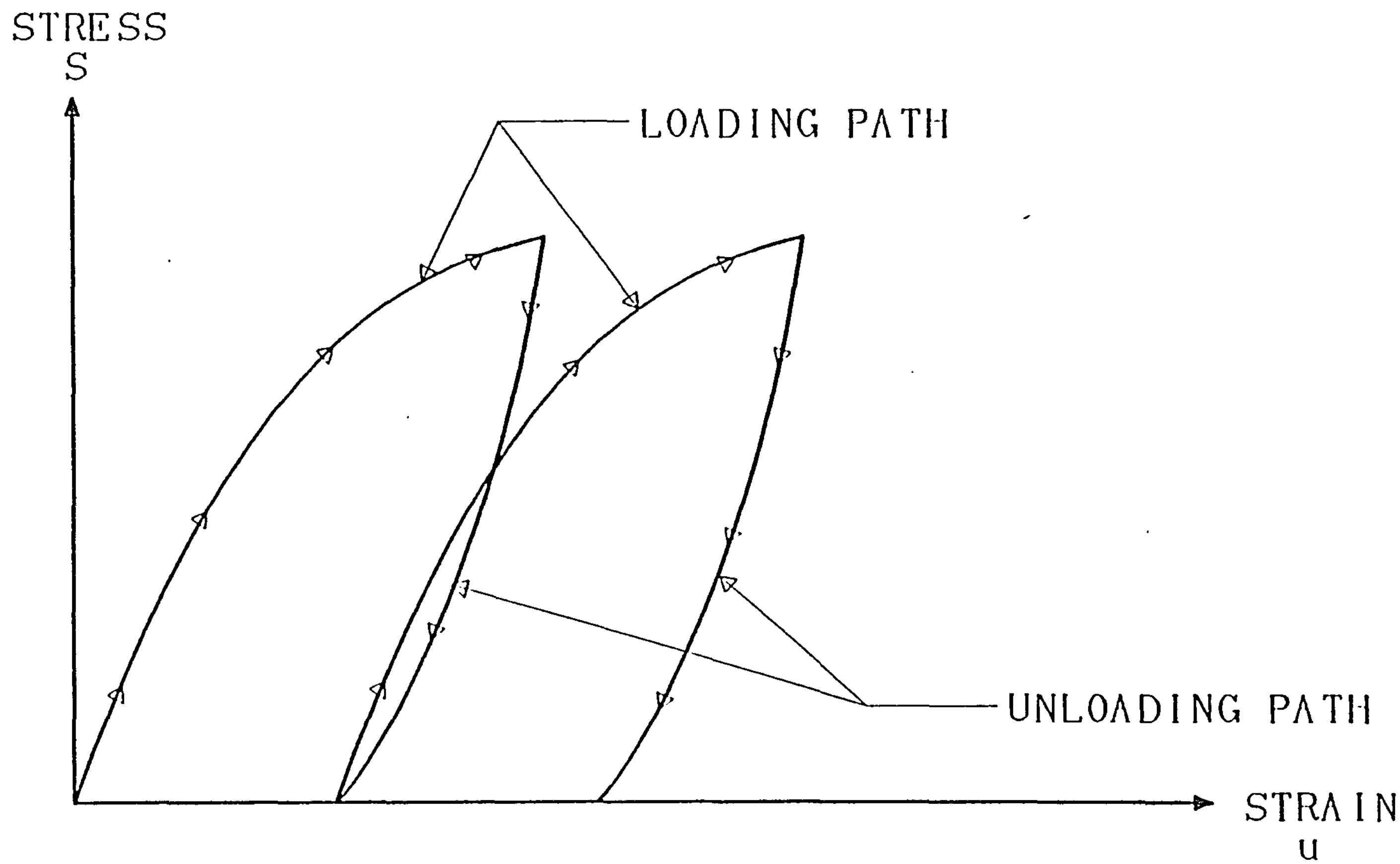


Figure 5. Loading and Unloading Path.

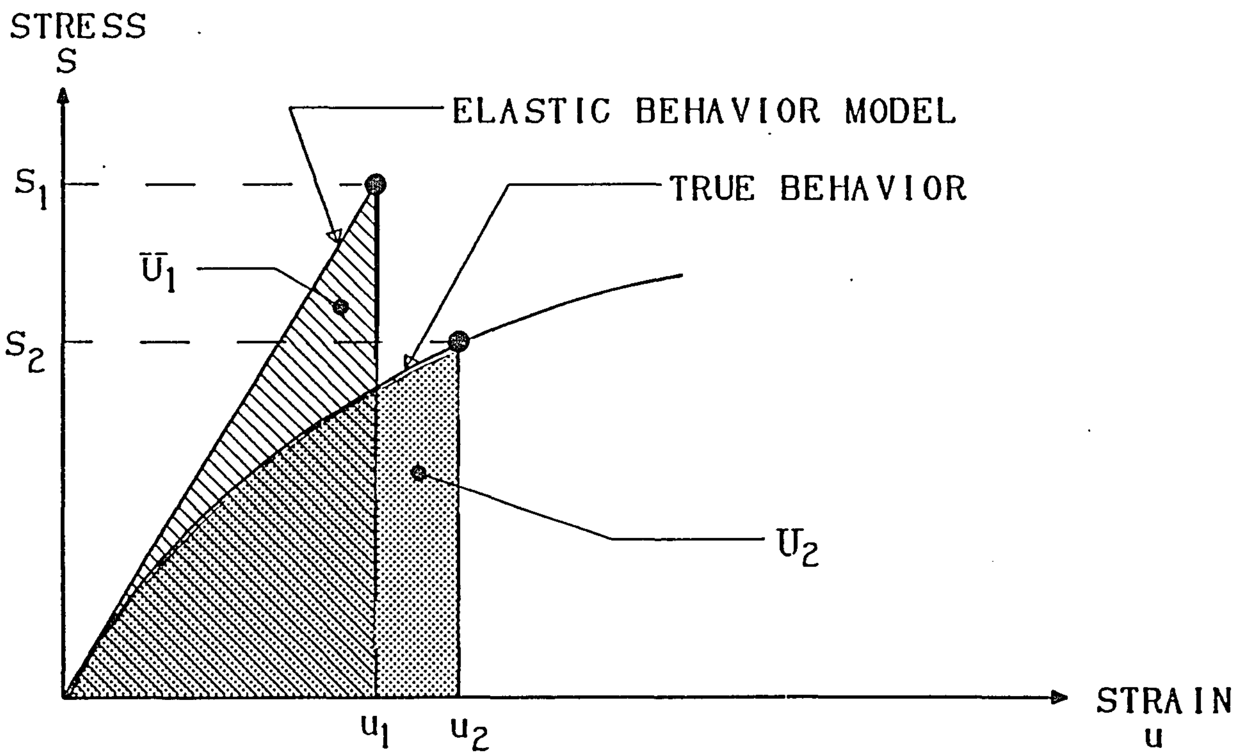
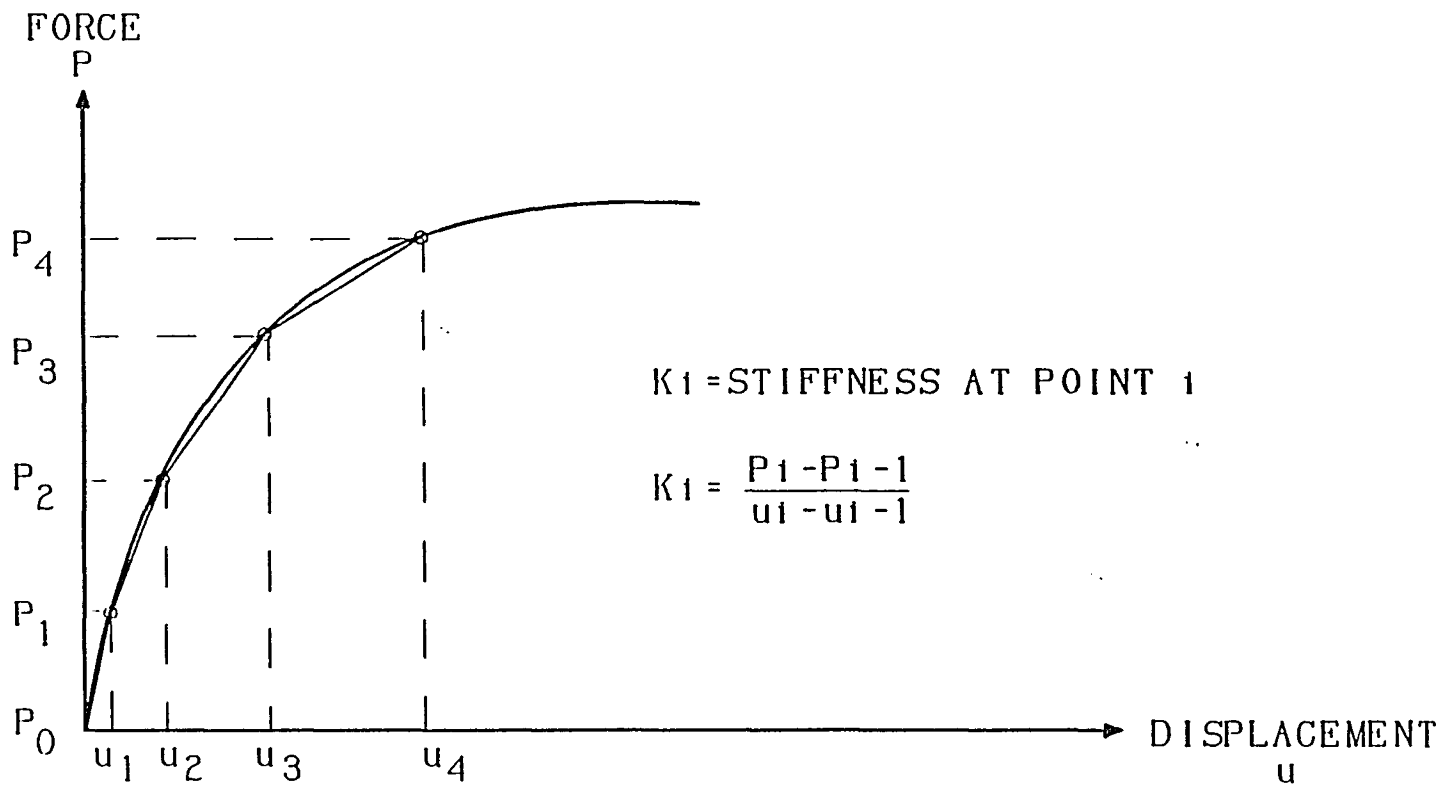


Figure 6. Equivalent Strain Energies U_1 and U_2.



<u>Figure 7.</u> Tangent Stiffness Method.

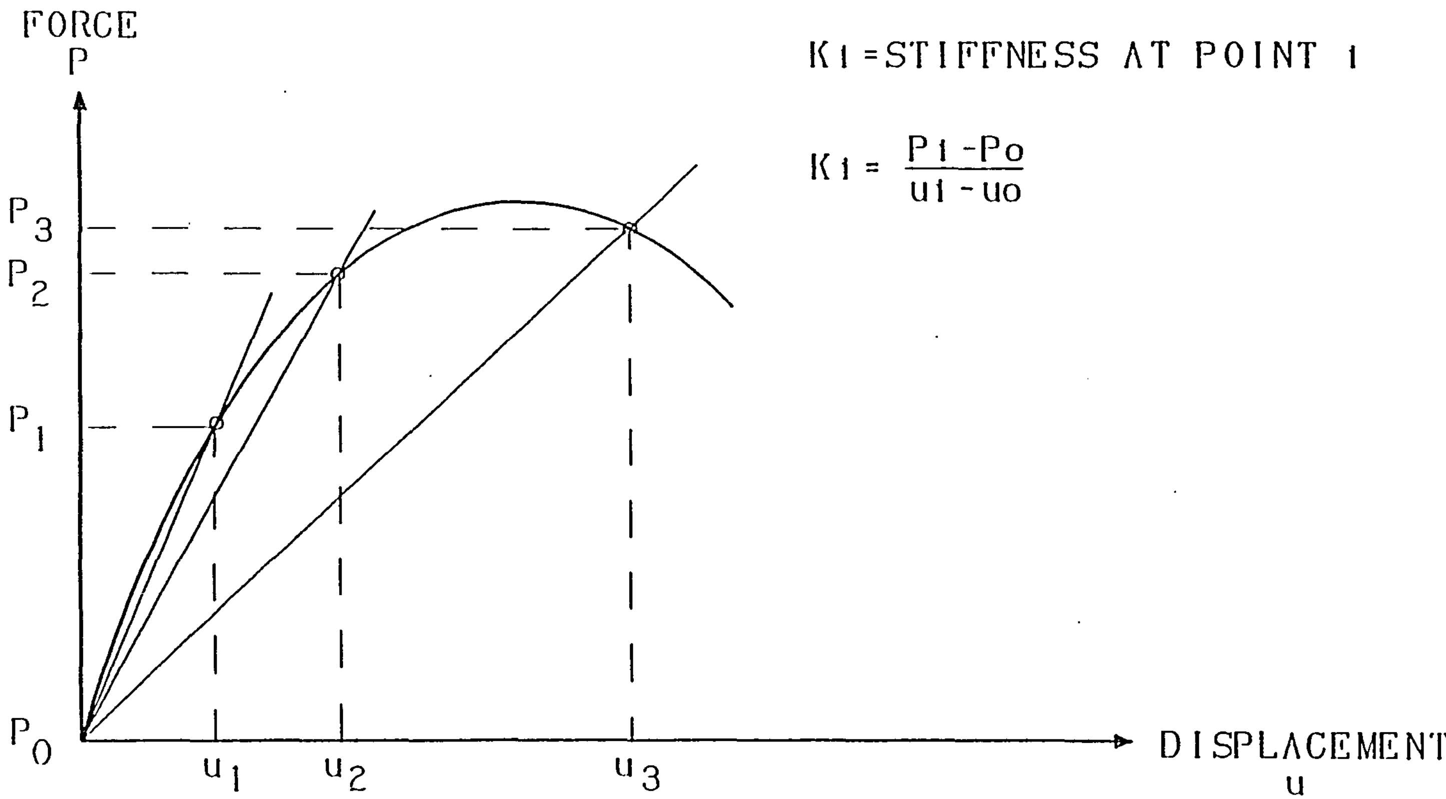


Figure 8. Secant Stiffness Method.

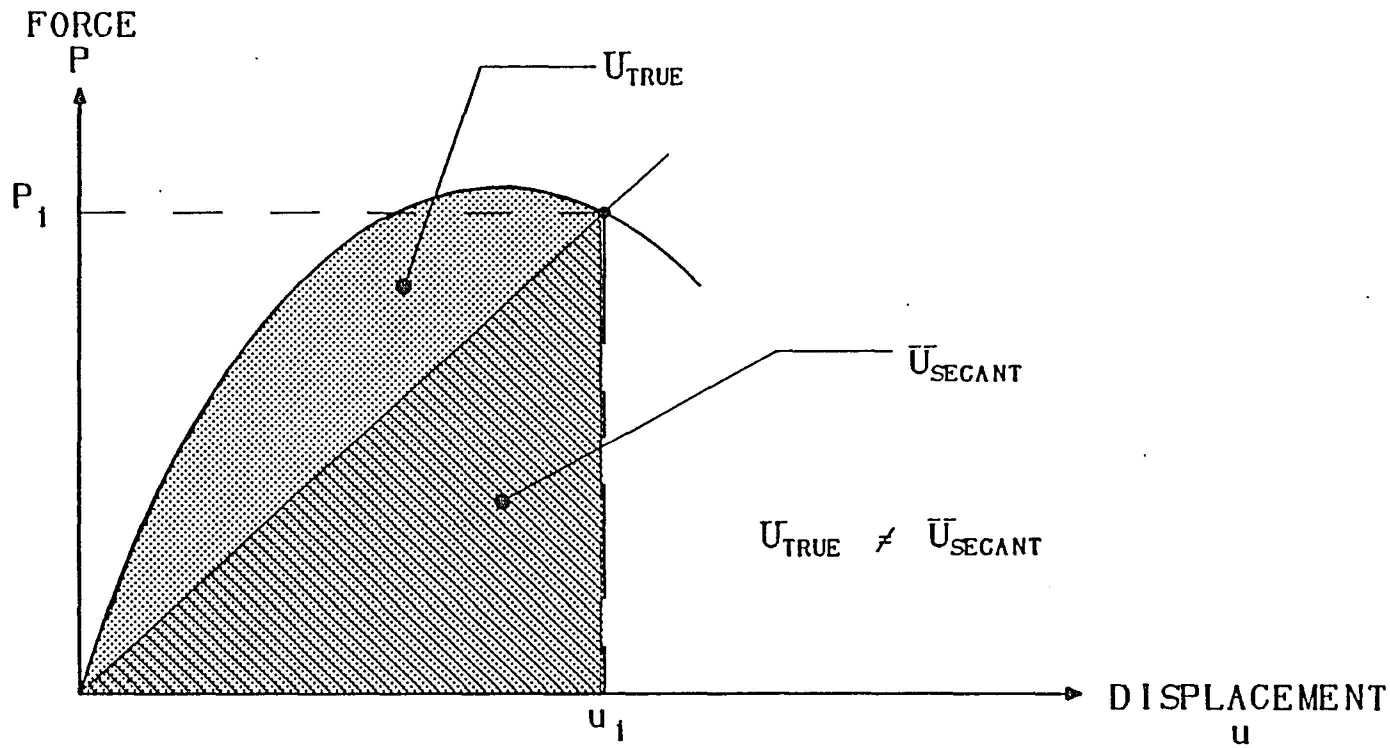


Figure 9. Strain Energy Violation of Secant Method.

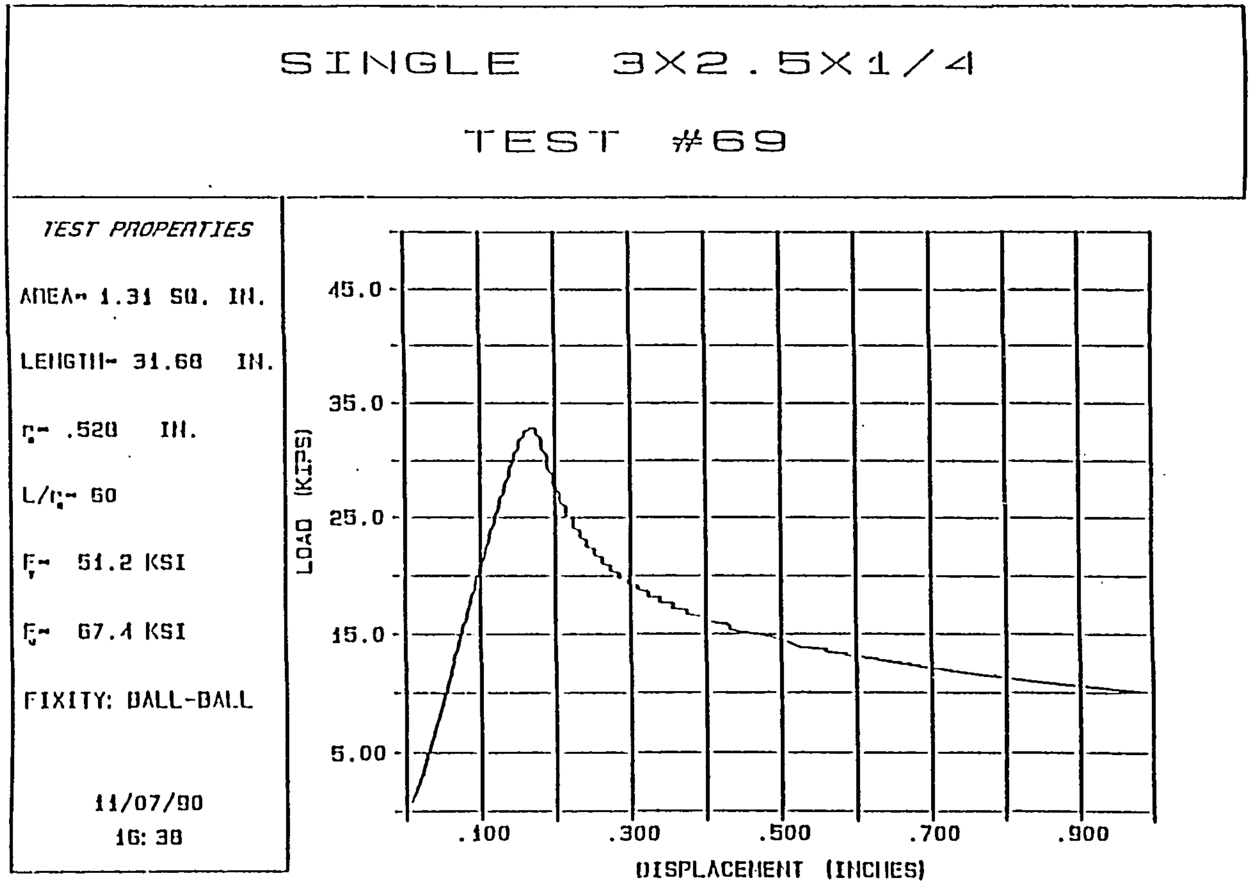


__Figure 10.__ Experimental Test Results - KL/R=60.
(Adapted from Bathon, P.128 (26))

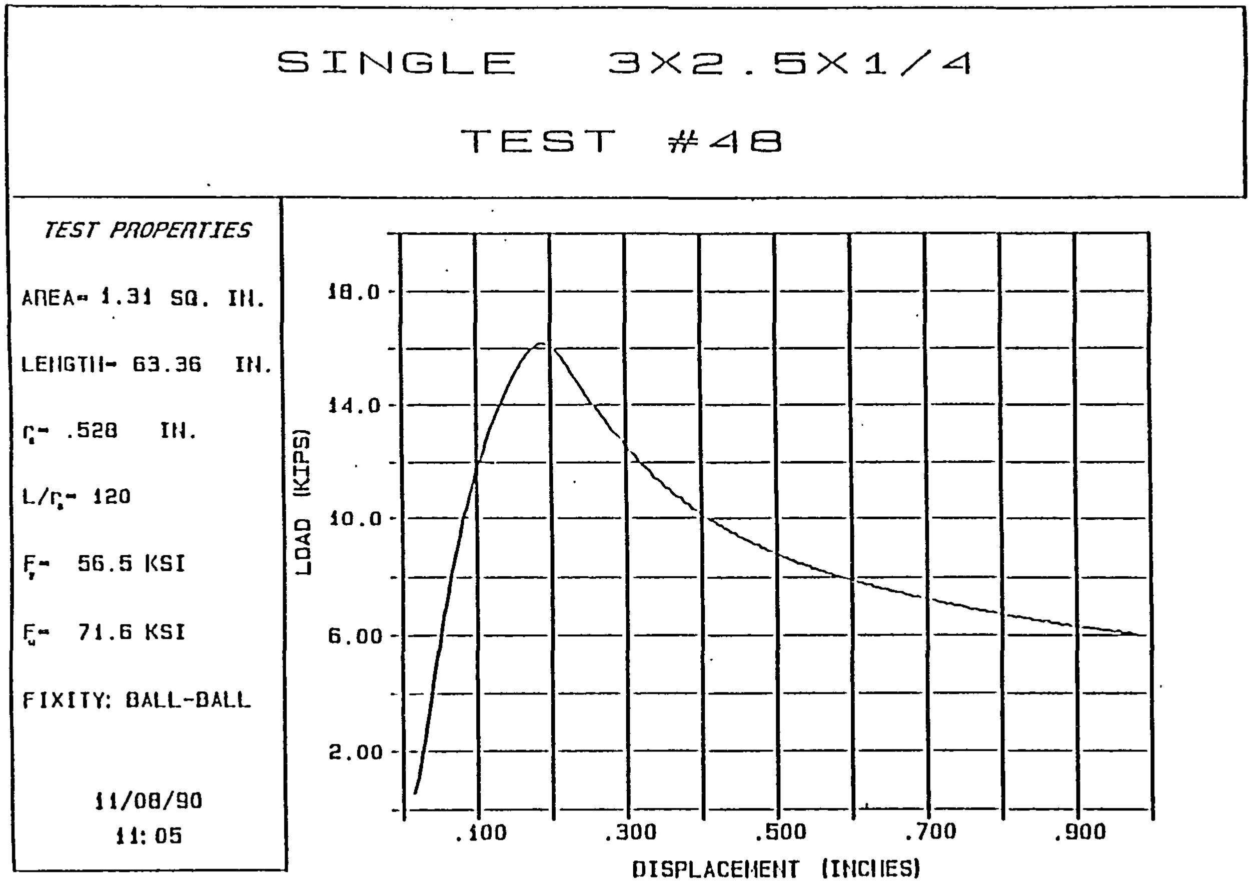
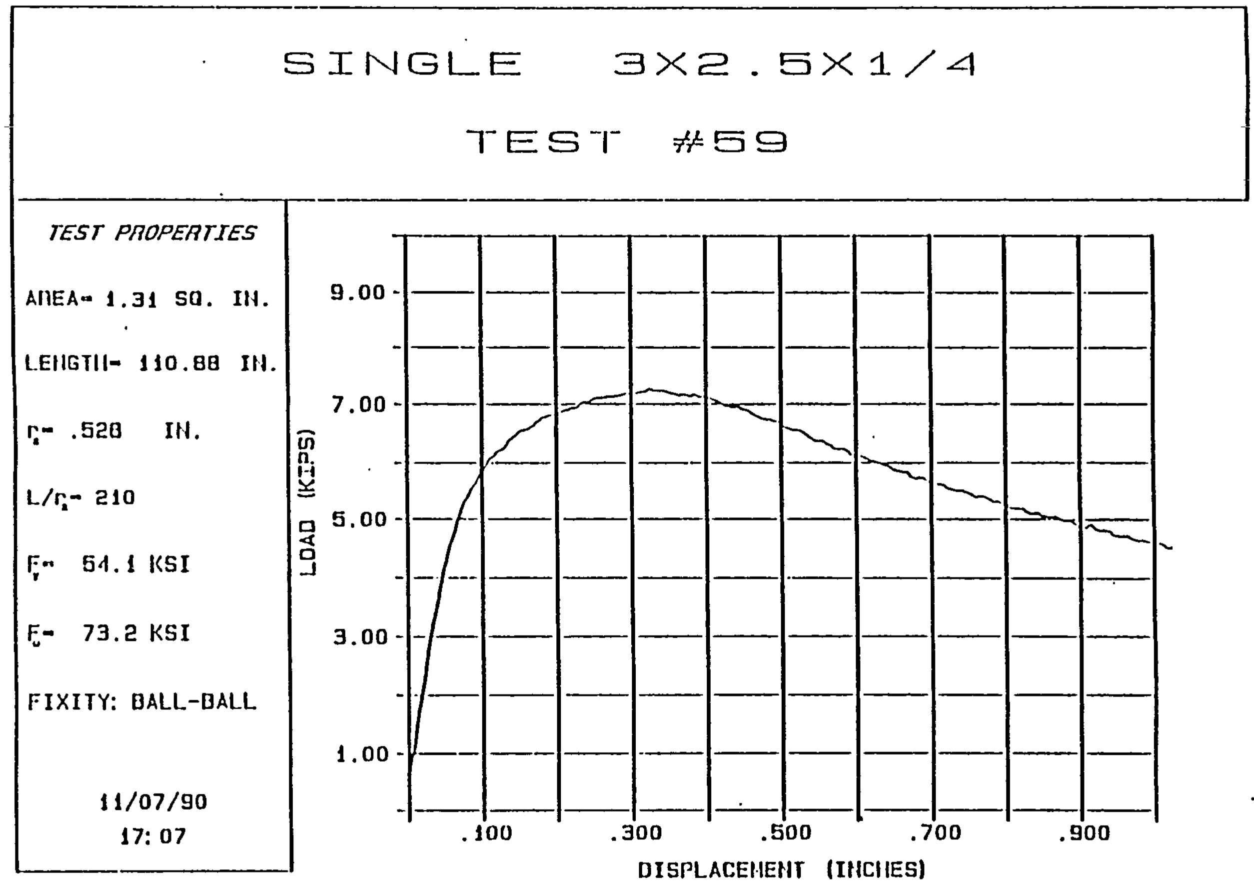


Figure 11. Experimental Test Results - KL/R=120.
(Adapted from Bathon, P.107 (26))



<u>Figure 12.</u> Experimental Test Results - KL/R=210. (Adapted from Bathon, P.118 (26))

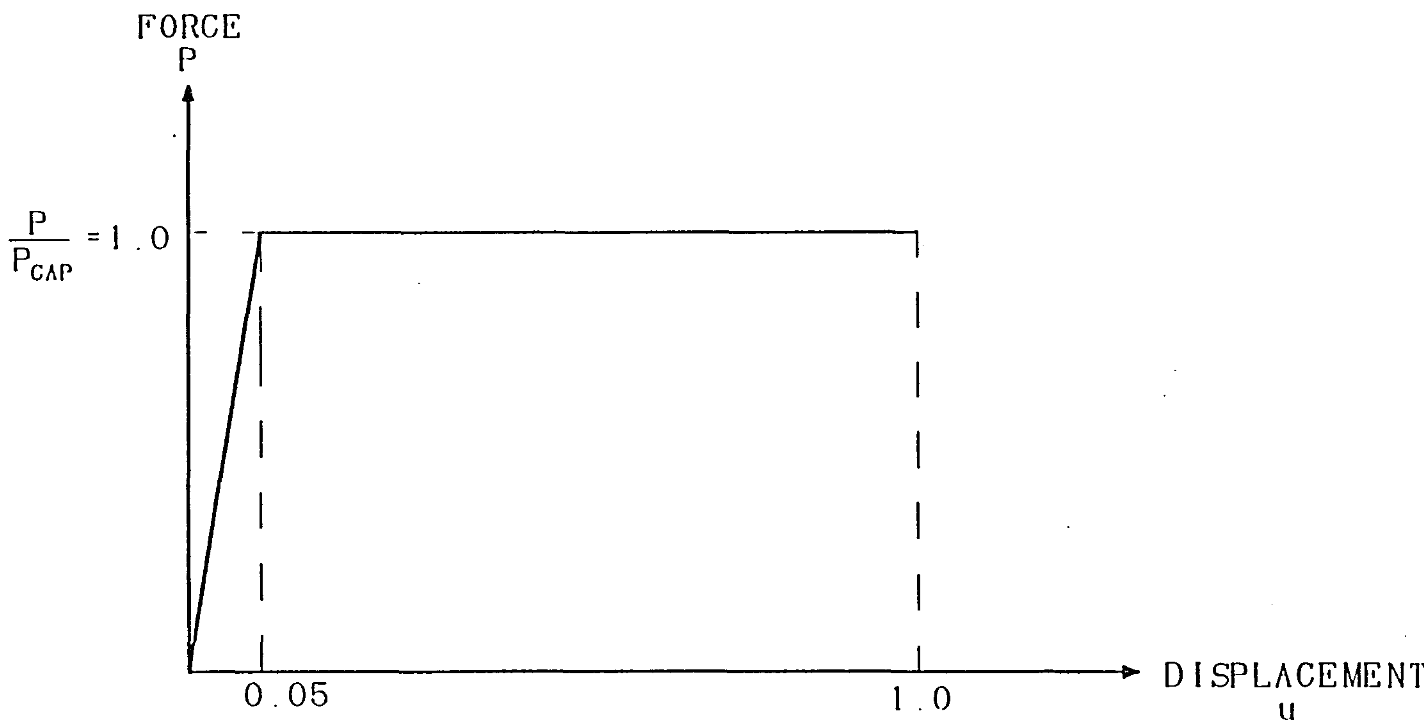


Figure 13. Normalized Bilinear Member Performance.

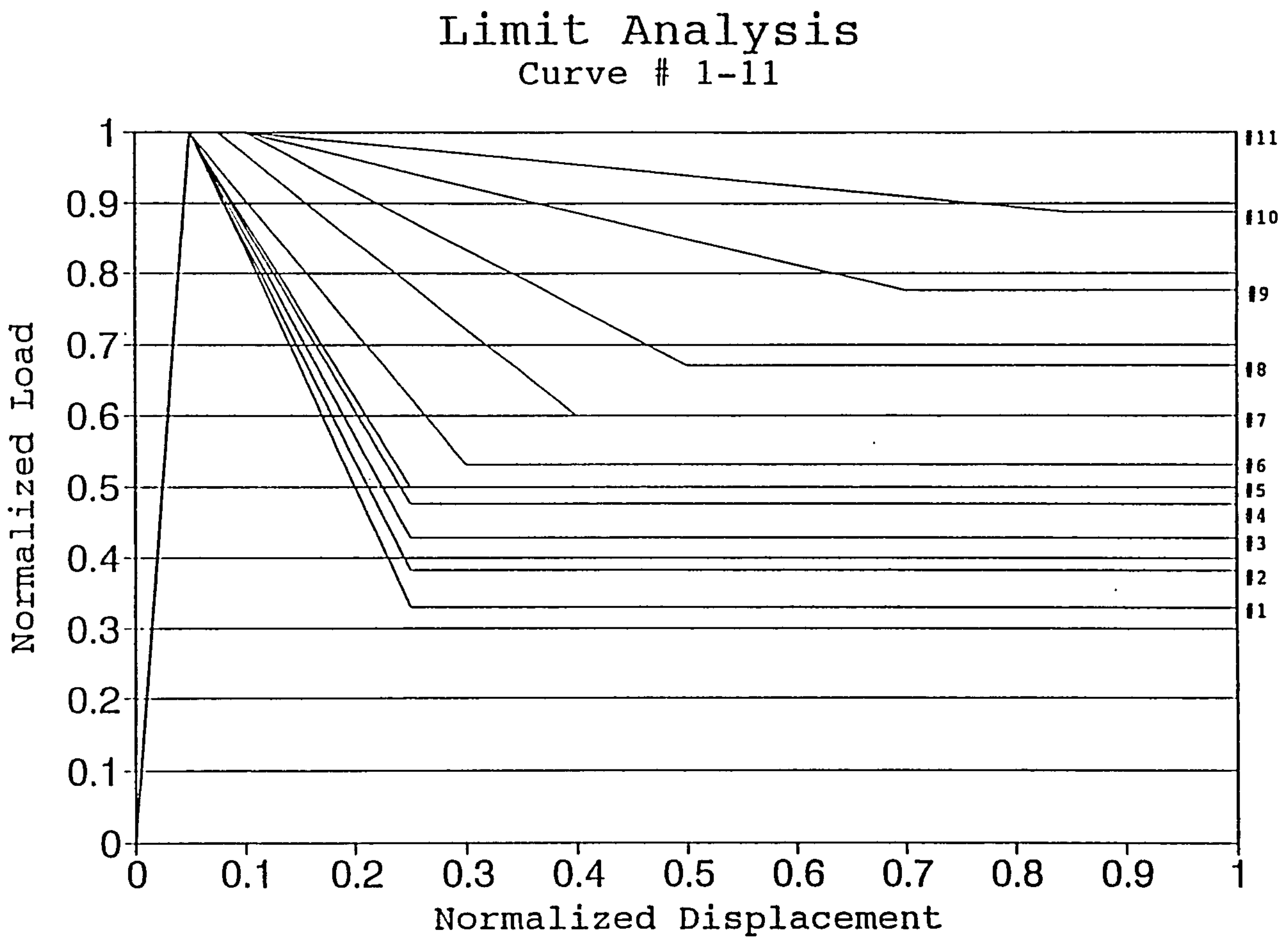


Figure 14. Normalized Member Performance Curves.

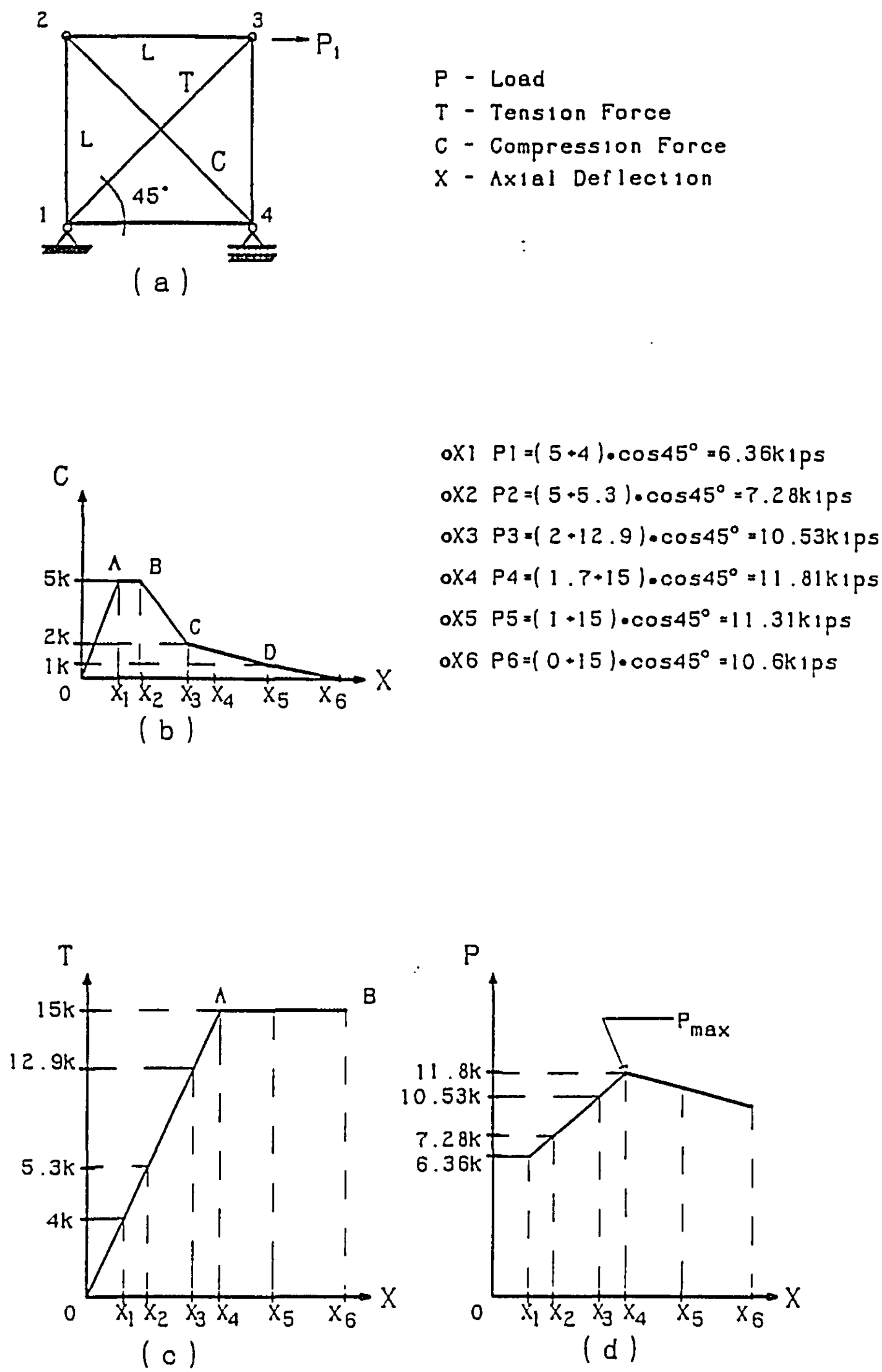


Figure 15. Simplified Structure.

CHAPTER II

LIMITATIONS OF CURRENT PRACTICE

DEGREE OF COMPLEXITY

As indicated previously, there are many factors which can complicate the successful completion of a nonlinear analysis. The complexity of a nonlinear analysis can usually be attributed to the nature of the iterative algorithm used in the program, the necessary data input, and the possibility of introducing modeling errors. This already evident complexity can further be aggravated, if the user who is supposed to perform the nonlinear analysis possesses no prior experience with the subject. Many frustrating hours can be spent by the user in an attempt to obtain a successful analysis, or worse, no results at all. The LIMIT B program is more complex to use than other traditional nonlinear analysis programs because it performs a 1st order nonlinear analysis that uses post-buckling member performances. Essentially, the constitutive laws used in the program are defined by experimentally determined member performances in terms of load versus displacement that account for the after failure performance of the individual members. Much expertise is required with the subject to perform a valid nonlinear analysis with the LIMIT B program.

An inexperienced analyst will have problems with any one, or all of the following concepts in a nonlinear analysis outlined in the paragraphs below:

A big challenge faced by a conventionally trained analyst in performing a limit states analysis is to extend his\her knowledge of the elastic analysis methodology to perform a nonlinear analysis. First, the analyst is required to extend elastic principles to understand the mechanics of a nonlinear analysis using bilinear member performances. In a nonlinear analysis using bilinear member performances (also called plastic analysis), failure is accepted and desired (30) in the form of plastic hinges as long as the stability of the overall structural system is maintained. Understanding the underlying mechanisms and how the internal forces are influenced is not intuitive to the novice analyst. Second, the analyst has to further extend these principles to the point where the analysis will use after failure member performances. At this point it becomes extremely difficult for the non-expert analyst to predict the underlying failure mechanisms, since the internal forces tend to redistribute throughout the structural system away from the potential failure mechanism (i.e. the load will flow from the failed member(s) to members that are able to carry additional load). It is therefore necessary for the analyst to not only know the state of the structure prior to and after the load application but also know all of the intermediate states. The previously discussed example of a limit states analysis shown in Figure 15 indicates how difficult it is to interpret the results of the analysis. The process of interpretation is complicated and varies for each structural system, and this typically confuses the novice user and sometimes even the expert.

There exists confusion among analysts about the definition of, and the differences between, the effective length (K*L) of the structural member and its true geometric length. Both of the parameters are used extensively throughout the LIMIT B, CURVEPLOT, and TOWER computer programs. Both parameters are used as data input for the above mentioned programs and have to be supplied by the user. The effective length of a structural member is defined as the largest portion (i.e. the greatest length) of the member between any two adjacent inflection points of the deflected shape. The true length of the structural member on the other hand is defined as the physical length of the member (i.e. in the finite element method it is the joint to joint length of the element), where the member is considered to be continuous. It is possible, that the effective length equals the true length (i.e. Euler Column with KL=L). It is also possible, however, that the true length may be a multiple of the effective length (i.e. in the case of intermediate bracing), or that the effective length may be a multiple of the true length (i.e. for the case of a cantilever column such as a flag pole KL=2*L). The effective length is used to calculate the compression capacity of the member in the nonlinear analysis. The effective length (i.e. more specifically the slenderness ratio KL/R) also determines the after failure performance of the member which ultimately influences the behavior of the overall structural system. One may conclude that the difference between the two lengths discussed above is not always readily apparent to the inexperienced user, and that errors can be made in the analysis and in the selection of the appropriate post-buckling member performance if these concepts are not clearly understood.

Within the CURVEPLOT program, the user is required to select the appropriate member performance curves to be used as input in the LIMIT B program for the nonlinear analysis. The member performance curve is defined as the graphically displayed, experimentally obtained, relationship between applied axial compressive loads and resulting axial displacements as a function of the geometric properties and the effective length of the member.

In order to select the appropriate nonlinear post-buckling member performance curve, the user has to perform a number of mathematical and graphical manipulations. Before the work described in this report, the user had to perform these manipulations to the curves manually. The original CURVEPLOT program does not adjust the nonlinear normalized model performance curves for the effects of the difference between the effective and joint to joint length of the member. However, it is necessary to denormalize the model curves appropriately based on the joint to joint length of the member. This will essentially stretch or shrink the member performance curves to appropriately model the experimentally obtained relationship between load and displacement. Furthermore, the model performance curve is denormalized and scaled to appropriately represent the theoretical compressive capacity of the member, which is a function of the effective length. Both of these manipulations eliminate the effects of bowing on the after failure performance of the member. The bowing effects are eliminated, because it is assumed that the member behaves perfectly elastic up to its compression capacity (i.e. the member is assumed to behave as an idealized two-force member). Essentially, the LIMIT B program only considers the after failure performance of the members. Depending on the magnitudes of these manipulations, the model performance curve will change its shape and performance. The last manipulation that is done consists of shifting the model performance curve towards the experimentally obtained member performance in an attempt to align the two curves. This shift eliminates the portion of the predicted post-buckling member performance that is associated with the bowing of the member and has to be done manually by the user. Bowing is usually caused by initial eccentricities of the applied load on a structural member loaded in compression. There is some ambiguity associated with these curve shifts (i.e. there exist multiple fundamentally sufficient answers, but usually only one answer which can be deemed most appropriate), which requires the user to possess some intimate knowledge with the nonlinear analysis method. The amount of shift used for the experimental data curve will ultimately determine the model behavior curve to be selected by the user. Since the results computed by the LIMIT B finite element program are highly sensitive to the selection of the model behavior curves, they are also equally sensitive to the appropriate shift of the experimental data curve.

Much experience of the user with the CURVEPLOT data interpretation, the data presentation, and the effects the decisions made will have on the computed results, is required, since these effects may render the analysis useless. However, even for the experienced LIMIT B and CURVEPLOT user it takes many trial and error attempts until an appropriate set of nonlinear post-buckling member performances is found for a particular problem. The majority of the problems with the LIMIT B program and the original CURVEPLOT database derived from the situation that no methodology existed (prior to this research) that would guide the user through the analysis. A nonlinear analysis algorithm was thus needed to alleviate the problems, and this required making changes to the LIMIT B and CURVEPLOT programs.

The original LIMIT B, as most other nonlinear finite element analysis programs, utilizes a specific convergence criterion based upon structural equilibrium to obtain a final solution. The convergence criterion is specified by the user as a percentage difference. The percentage constitutes the difference between the member load calculated using the conventional direct stiffness method, and the member load extracted from the post-buckling member performance curve as a function of the calculated axial displacement. Depending on the shape of the member performance model behavior curve, the convergence criterion applied to the loads directly affects the results computed by the LIMIT B program. This phenomena is a result of the characteristics of the member performance model curve. If the model curve of a member displays a flat post-buckling slope, the convergence criterion imposed on the load can be fulfilled for a larger range of deflections than if the post-buckling slope of the curve is very steep. If the axial displacement of the member calculated by the direct stiffness method falls within the range of possible displacements determined from the convergence criterion on the loads, the probability is high that the results computed by LIMIT B are valid. On the other hand, if the axial displacements calculated by the conventional method do not fall within that range, it may cause the results computed to be inaccurate. Again, as mentioned previously, it is necessary for the user to have enough experience with the nonlinear finite element analysis method and the LIMIT B program to assess the effects of the convergence criterion.

A number of other problems may arise during the selection of the system control data for the original LIMIT B analysis. A variety of control parameters have to be specified by the user to guide the analysis towards a successful solution.

The user has to specify the maximum number of trial stiffnesses for a given load level for which the program will attempt to obtain a solution. If the number is too low, the program may not be able to find any solution, or the solution found by the program may be not valid. Either one of the outcomes may prove to be frustrating to the user, since it is not always obvious to the inexperienced person, that a problem exists with the analysis. On the contrary, if the number of attempts to obtain a solution is too high, a number of problems may arise during the analysis. It is possible, that the LIMIT B program will be able to find a solution that lies too far out on the post-buckling member performance making the collapse load factor calculated useless. It is also possible, that LIMIT B would calculate a solution for the collapse load factor using an inappropriate set of member performance curves. If the maximum number of trial stiffnesses for any given load factor is selected inappropriately, any solution can be calculated (i.e. the results computed by the program will vary every time the analysis is run). Again, it is necessary for the user to be an expert in using the LIMIT B program, since the appropriate selection of the control parameter is influenced by the type of structure that is analyzed.

The user is also required to specify the total maximum number of trial stiffnesses the program will attempt for all the load levels during a complete run. Even though the limitation will protect the user from a runaway solution, it may also create some problems. If the number is too small, the program will not be able to find a valid solution, or may even shut down before any solution at all is obtained. Much intuition and expertise is required from the user to select the appropriate system control data, since "low" and "high" are relative quantities which change depending on the problem context (i.e. the selection of the most appropriate set of parameters is unclear, since appropriate values are relative and context dependent).

Some of the more conventional problems associated with a nonlinear analysis, such as modeling errors, undetectable errors, and the lack of consistency in the results will be discussed within the subsequent sections. Despite all of the problems, the results obtained from LIMIT B can be considered a valuable tool in the analysis and design process, if the analysis is performed by an experienced user who is aware of all of the consequences. However, because of the intuitive nature of the analysis, even an expert will often make mistakes. An expert system application acting in the role of an independent consultant would produce more consistent and more realistic results.

MODELING ERRORS

In addition to the problems previously mentioned, there also exist a variety of other problems that can be associated with the model chosen to represent the structural system in the nonlinear analysis. Most of these errors may also occur during a conventional elastic analysis, and are therefore not necessarily unknown to the average designer. Some of the problems listed below though, are specific to a nonlinear analysis with LIMIT B, and are not known to the novice user but are usually caught by the expert analyst. The most common modeling errors are:

- Connectivity Problems (i. e. problems with the model definition. These are problems as a result of members which are connected to joints which do not exist, members that are connected to the wrong joint, members that are connected to only one joint, etc.)

- Joint Labeling & Coordinates (i.e. this is usually caused by complicated joint generation schemes, and not normally readily apparent. Most connectivity problems are usually caused by incorrect joint labeling and misplaced coordinates.)

- Member Information (i.e. to specify incorrect member geometric section properties can cause a variety of problems. These problems include instabilities, very small or excessive deformations, and inaccurate stiffnesses and member loads.)

- Specified Deflections (i. e. incorrectly selected specified deflections, and\or missing specified deflections. These problems may cause inaccurate results and global instabilities. Again, it may not be readily apparent to the user that this condition exists.)

- Load Application & Magnitude (i. e. loads may be placed in an inappropriate alignment with the local or global coordinate system. In addition, loads may be applied to joints which may prove to be unstable if loaded in the specified direction or the wrong load component may be flagged to have incremental loads added. Furthermore, it may have loads added in increments by a factor which differs from the number specified by the user.)

- Curve Selection (i. e. inappropriate performance curve selection may, as previously indicated, render the analysis inaccurate or even completely useless to the engineer. In addition, the CURVEPLOT database contains only information that was derived from experimental tests of members with a slenderness ratio in the range of 200 to 350 (i.e. long members). The user is required to extrapolate from the experimental data in order to model short and intermediate members (i.e. members with a slenderness ratio in the range of 0 to 180), the effects of eccentrically applied loads, differences between the performance of single and double angle members, strength variations, and lap and butt joints. Furthermore, a total of only eleven member performance model curves is contained in the CURVEPLOT database. Based on experience, it may be stated that this set is not sufficient to adequately model all of the members that are normally used in the design of a lattice type structure. Much experience is needed on the part of the user of the LIMIT B program to compensate for these shortcomings and to assess the influence of these factors on the calculated results.)

- Support Conditions (i.e. typically, in the case of lattice type structures, it is not known to the analyst where the structure will be located. Currently, most elastic or inelastic analyses of lattice type structures do not take into account the effects of the site specific soil response. No matter how refined the analysis of the structure is, the analyst always has to verify the design by including the effects of soil response. However, the consideration of elastic or inelastic site specific soil response is out of the scope of this particular research and development project.)

- Program Control (i.e. the program and its solution algorithm are very sensitive to the control parameters. A bad selection of control parameters may create a variety of frustrating runtime problems.)

Most of the common problems mentioned above can be avoided by the expert analyst and should not interfere with the successful completion of a nonlinear analysis, as long as attention is paid during the preparation of the data input file. Nevertheless, there is always a chance, that even the experienced analyst will make a mistake during the preparation of the model and the creation of the data input file. However, if this occurs, the experienced user will be more successful in the elimination of the cause than the inexperienced user. Extensive error trapping included in the analysis program can avoid these problems.

HIDDEN AND NUMERICAL ALGORITHM RELATED ERRORS

Another class of problems which may be encountered during a nonlinear analysis are program specific and may or may not be transparent to the user. These problems usually are directly related to the numerical algorithms and solution schemes used within the analysis program. This type of problem may at times be hard to detect, and even harder to eliminate, since the source of the problem will usually remain obscured within the program code. The detection of these problems requires the user to be an expert in nonlinear analysis methods or to be assisted by a program that can detect these problems and provide advice. The Limit States Analysis Module that has been developed during the course of this research assists the user in the discovery of these types of problems. Some of the common problems within this class are:

- Extraneous Joints (i. e. a joint with two or less members attached to it, such as a joint accidentally placed at the midspan of a continuous member. It can also be a joint which possesses no stiffness in one of the three orthogonal directions.)

- Instabilities (i. e. a joint that has multiple members connected to it in only two of the three orthogonal directions. Essentially a joint with multiple members attached to it where the summation of the member stiffnesses equates to zero in one of the three orthogonal directions.)

- Runaway Solution (i. e. the majority of nonlinear analysis codes utilize some kind of iteration scheme to obtain subsequent improved approximations to the final solution. In certain situations, it is possible for the numerical iteration scheme to become unstable and diverge. As a result, the program will not be able to zero in on the final solution and thus continues the iteration scheme indefinitely.)

- Automatic Restraints (i. e. some analysis programs such as LIMIT B, offer the user the option of having artificial spring restraints automatically applied by the program to all the joints which show an out-of-plane instability. Even though the option is usually beneficial, it may create a problem if the instability is large, as a result of an invalid model, terminating the analysis with a computed structural capacity below the true capacity.)

INTERPRETATION OF RESULTS

Upon completion of a successful nonlinear analysis using the LIMIT B program, the user will obtain a data output file displaying the results. These results will have to be reviewed and interpreted by the user in order to draw the correct conclusions about the strength and behavior of the structural system, or to determine the next step of refinement necessary to improve the solution. However, the appropriate interpretation of the results requires an intimate knowledge of the analysis program, the post-buckling performance of members, the way the CURVEPLOT database should be used, and the nonlinear finite element method. As a direct result of these requirements, the average user will experience difficulties with the interpretation of the results and will be confused about the next appropriate level of refinement to be used in the analysis.

The inexperienced user will spend too many hours on the review of irrelevant data, or in the attempt to determine the failure mechanism of the structural system. The waste of time will result in an inefficient operation and a decreased productivity on the part of the user. An evaluation module, which would guide the user through the interpretation process, could provide invaluable help to the user, and at the same time increase productivity.

HUMAN FACTORS AND RESOURCES

As it was previously stated, the successful completion of a LIMIT B run requires the user to be an expert in the field of nonlinear analysis and to be familiar with the finite element analysis program. However, the average user (such as the typical college educated engineer) will not have been exposed in detail to the nonlinear analysis concept during the course of his/her formal education, since the subject is usually taught in the latter part of graduate studies. In addition, it is not likely, that the average engineer will be exposed formally to the subject during his/her professional career.

Nevertheless, due to unforseen circumstances, such as a shortage of resident experts or the development of new design and analysis specifications such as reliability based design and analysis (RBD), the engineer may be required at some time during his/her career to perform a nonlinear analysis. Clearly, this will pose a problem, since the analysis performed by the engineer could become a long, frustrating task, and will probably end up being inaccurate. At the least, assuming that the mistakes made will be caught by the resident expert, the company and the engineer will have lost a considerable amount of time, resulting in a decrease in productivity and much frustration. This outcome is contrary to the purpose of the nonlinear analysis, which is the improvement of the cost effectiveness of the design process without an increase in the demands on the human resources. The cost competitiveness of the market will therefore create a dilemma for the company which can not easily be resolved by traditional means.

One of the possible, economically feasible solutions to the dilemma for the company is to develop a program which will guide and train the inexperienced engineer in the nonlinear analysis method, without introducing an additional strain on the work load of the resident human expert. Guidance and training will be provided, if the program will act in the role of an advisor to the user, suggesting related information, proposing additional alternatives, providing recommendations in unusual situations, and suggesting possible causes of actions.

CHAPTER III

RESEARCH METHODOLOGY AND PROCESS

SOLUTION TO CURRENT LIMITATIONS

The development of an expert system application together with the improved solution algorithm would resolve any of the shortcomings and problems associated with a nonlinear analysis using post-buckling member performance curves described in Chapter II. This application would help the user throughout all phases of the analysis. It was anticipated, that it would be necessary to develop an analysis package that would combine the knowledge and experiences of two experts in the field of nonlinear analysis using post-buckling member performances with the LIMIT B and CURVEPLOT program modules. The idea was to combine the knowledge of two human experts with the existing program modules to develop a tool that could assist the analyst in a production environment.

A feasibility study was performed by the author to determine what tests were necessary to be done to develop this complete nonlinear analysis package, which was named the Limit States Analysis Module. Upon completion of the study, a plan was developed that describes the steps that are deemed necessary to develop the Limit States Analysis Module. The research and development tasks that are included in the original plan are described below:

- An expert system application development shell will be selected. The selection of the development shell will depend to a large extent on the requirements imposed by the particular problem context. A detailed discussion of the advantages and disadvantages of the development shells can be found in Appendix B.

- The knowledge base that will be used in the Limit States Analysis Module will be solicited from two experts in the field of nonlinear analysis. The results will be used to structure and program the combined knowledge. The human expertise will be verified and tested before it will be incorporated into the Limit States Analysis Module program.

- The original LIMIT B program will be changed to incorporate hardware and software advances in the microprocessor industry. In addition, changes will be made to the format of the input and output files that will be advantageous to the development of the Limit States Analysis Module.

- A new nonlinear analysis algorithm for the LIMIT B program using post-buckling member performance curves will be developed. The new algorithm will reduce the time required to perform a nonlinear analysis with the LIMIT B program using post-buckling member performance curves and improve the consistency of the results.

- The post - buckling member performance database contained in the CURVEPLOT program will be expanded by experimentally testing steel angle members. A test program will be developed that will benefit the development of the Limit States Analysis Module.

- The post - buckling member performance database contained in the CURVEPLOT program will be changed to include the new experimental data obtained from the physical member tests that will be performed at Portland State University.

- New post - buckling member performance model curves will be developed to reflect any new information discovered from the experimental member tests.

- A data input file numerical pre-processor will be developed that will check the data input file for format, inconsistencies, basic modeling errors, and permissible data ranges.

- A graphical pre-processor will be developed that will allow the user to view the structural model prior to the analysis. Options will be added to the pre-processor as a need is recognized during the course of the research project.

- A graphical post-processor will be developed that will allow the user to view the structural model, view the critical members, display post-buckling member performance curves, and see the deflected shape of the structure upon completion of the analysis. Other options will be added as a need is recognized during the course of the research project.

- User friendly screen editors will be developed that will allow the user to change any of the input data and control parameters. The screen editors will feature on-line help screens that should be available for any of the required input values.

- Develop friendly graphical user interfaces (GUIs) that allow the user to activate any of the components of the Limit States Analysis Module.

- Develop a plan to verify and validate the complete Limit States Analysis Module.

- Debug and verify each of the Limit States Analysis Module components, in particular file reads and writes, accuracy of solution, and the accuracy, usefulness, and appropriateness of the results obtained.

- Validate the Limit States Analysis Module by having inexperienced analysts use the program to calculate results which will then be compared to results obtained by full scale tower load testing.

- Change the LIMIT B user manual to reflect the changes that will be made to the input and output structure and the analysis algorithm of the LIMIT B.

- Develop a configuration sheet, including hardware and software requirements, and installation procedures for the final version of the Limit States Analysis Module.

DEVELOPMENT OF THE LIMIT STATES ANALYSIS MODULE

In the following paragraphs a description will be presented of the individual steps that were taken to develop the Limit States Analysis Module. The section will mention some of the problems that were encountered throughout the research and development phase, present the solutions that were devised to solve the problems, and discuss advantages and disadvantages of the decisions made during the development of the Limit States Analysis Module.

As a first step, a number of literature searches were performed. Subject areas consulted included topics such as reviewing information on general expert system methodology, methods and principles typically used in the development of expert system applications, existing expert system applications in the field of structural analysis and mechanics, and becoming familiarized with commercially available development software packages. Furthermore, information was obtained about the theory and principles of the limit state analysis methodology, existing limit state analysis applications, and the development and use of post-buckling member performance curves. Finally, related information in the field of nonlinear analysis was reviewed and evaluated, and in particular with respect to existing nonlinear analysis solution algorithms. A large portion of the information that was gathered in the various literature reviews was utilized in the development of the Limit States Analysis Module.

The next step that was taken consisted of the selection of the most appropriate development shell to be used in the development of the Limit States Analysis Module. The selection process was made somewhat difficult, since the evaluation of the programs had to be based on information taken out of product literature. It was not possible for the author to actually physically evaluate these software packages, since it was financially impossible to purchase more than one of the software programs. Upon completion of the evaluation process described previously, it was decided that the NEXPERT OBJECT development shell was best suited for use in the development of the Limit States Analysis Module. At this point, the development of the program was only in the conceptual stage, and NEXPERT OBJECT was considered the most versatile of the development tools reviewed.

Once the NEXPERT OBJECT program was purchased, the author proceeded to familiarize himself with the software package. The program's development interface (i.e. the rule editor, class editor, object editor, metaslot editor, etc.), the inference engine processing routine, the database interface, the runtime definition language, the report language, and the user interface design module were studied in depth. The process of familiarization proceeded until all of the capabilities of the program were assessed.

A functional prototype of the Limit States Analysis Module was conceptualized and developed. The functional prototype attempted to assess all options of the NEXPERT OBJECT software package and represented essentially a feasibility study of what could be done using the development tool. It was deemed necessary to determine the limits of the development tool prior to the conceptualization of the final version of the Limit States Analysis Module. The functional prototype did not contain any of the actual knowledge base. However, the prototype did contain a working user interface, read and write modules, and a simplified version of the on-line help. A variety of shortcomings and problems were discovered during the development of the prototype which made it necessary to change the role originally anticipated for the NEXPERT OBJECT program. A detailed discussion of the problems encountered and the solutions devised is presented in one of the following sections entitled "Changing the role of NEXPERT OBJECT".

Next, it was deemed necessary to study the existing LIMIT B nonlinear analysis program. Example problems were analyzed to determine how the original LIMIT B program functions, how a typical nonlinear analysis using post-buckling member performance curves is performed, and to assess the limits of the program and the problems that can be encountered in a nonlinear analysis. A number of problems were discovered with the LIMIT B program which were deemed significant enough to necessitate some changes to the analysis code. A detailed description of the problems encountered and how these problems were eliminated is presented in one of the following sections entitled "Modification of the LIMIT analysis program".

At the same time, since it is a required input of the LIMIT B program, CURVEPLOT was examined in much the same manner. The CURVEPLOT source code was inspected, the database was evaluated, and the shortcomings of the program were assessed and documented.

The author decided that it was imperative to study the two programs in detail, since he was responsible for soliciting and encoding the expert knowledge associated with a nonlinear analysis that uses post-buckling member performance curves. The majority of the questions and experiences used in the interviews with the experts were based on the findings of this particular investigation. The idea was essentially to have an inexperienced analysts (i.e. the author) perform the nonlinear analysis to develop exactly the types of questions that were necessary to be asked from the experts. In this way, it was anticipated that the knowledge base solicited from the experts would be appropriate and complete for the development of the Limit States Analysis Module (i.e. the questions would be focused on the portions of the analysis that create problems for the inexperienced user).

The questions, problems, and shortcomings formulated during the previously described process were investigated in a series of interviews with the experts. Many iterations were necessary until a complete picture of the knowledge associated with the LIMIT B nonlinear analysis using the CURVEPLOT post-buckling member performance database emerged. In addition, general shortcomings of the nonlinear analysis process were assessed such as described in the preceding chapter. A more detailed description of the methods that were used to solicit the knowledge base is presented in Chapter IV.

One of the biggest problems that was encountered within the preliminary phase of the knowledge acquisition process was that the database of the CURVEPLOT program did not contain any experimental test results of intermediate and short compression members. Short and intermediate member behavior had to be therefore extrapolated from test results of long compression members. The database also did not contain any information that could be used to describe the post-buckling member performance behavior of double angles. This situation was clearly not adequate and had to be resolved. It was decided to expand the post-buckling member performance database by performing physical member tests for short and intermediate compression members, for both single and double angles, which would then supplement the existing CURVEPLOT database. A test program was developed and refined and its appropriateness validated. A detailed description of the decisions made and the problems encountered during the development of the test program and the actual member testing is presented in the section entitled "Expansion of the Post-Buckling Member Performance Database".

A parametric study of the LIMIT B program was performed by the author in order to assess the sensitivity of the nonlinear analysis to changes in the control parameters. It was necessary to determine how each individual control parameter influenced the results produced by the LIMIT B nonlinear analysis program. A parametric study had been performed previously, but the results had never been recorded and evaluated with respect to the development of a knowledge base. The findings discovered in the parametric study were then utilized to supplement the expertise of the experts and to fill some of the gaps in the solicited knowledge base.

Finally, it became necessary to conceptualize the final design of the Limit States Analysis Module. A functional description was developed that defined what components were necessary to be included in the Limit States Analysis Module to allow an inexperienced analyst to perform a nonlinear LIMIT B analysis using post-buckling member performance curves. Early on it was decided to divide the program into individual components to allow each of the components to be verified and\or modified without effecting the performance of any of the other program modules. Employing this system would speed the development process since it would make the verification of each component more convenient.

The final design of the Limit States Analysis Module was based on the information contained in the knowledge base. A detailed description of the final modular design is presented in the section entitled "Development of the Limit States Analysis Module". The decision was made to divide the Limit States Analysis Module into individual components. The more significant components are listed below:

- Control Program
- Curveplot Graphics Module
- Loadhistory Graphics Module
- Towerplot Graphics Module
- Bandwidth Optimization Module
- Pre-Processor
- Expert Input Evaluation
- Expert Output Evaluation
- User Interface

The circumstances that led to the development of the various graphic modules and the problems that were encountered are described in detail in the latter sections of this chapter. The development of the pre-processor, the expert input evaluation, and the expert output evaluation is discussed in detail in Chapter IV.

The last task that was performed in the development phase of the Limit States Analysis Module consisted of the creation of an extensive verification and validation program. Due to the nature of the solution algorithm, the program had to be validated in addition to the typical verification process. A detailed description of the verification and validation process that was developed for the Limit States Analysis Module is presented in Chapter IV.

CHANGING THE ROLE OF NEXPERT OBJECT

A number of shortcomings and problems with the NEXPERT OBJECT software program were discovered during the development of the Limit States Analysis Module prototype. A description of the more significant problems that were encountered during the development phase of the prototype is presented in the following paragraphs. It was found out later, that many of the shortcomings described below were eliminated in the most recent version of the NEXPERT OBJECT program. However, the latest version of the program was not released until after the Limit States Analysis Module had been completed.

NEXPERT OBJECT is able to display graphics such as bitmaps within the developers interface which is run as a MS Windows application by utilizing its proprietary "show" function on either the left or right hand side of a rule. However, NEXPERT OBJECT is not able to display any graphic bitmaps once the knowledge base is compiled and run as a standalone executable application. This problem exists because the standalone application runs only in a character based screen mode while the developer's interface in which the application is created utilizes a GUI.

The NEXPERT OBJECT development interface is graphics based and runs as an application within the MS Windows environment. The user interface development tool is also graphics based and runs as an application of the MS Windows environment. The user interface development tool is used by the programmer to design and program the menus and screens that the user will see if the final standalone application is run. However, despite the fact that the user interface is designed in a graphical environment, the final runtime user interface is character based only (i.e. any of the components of the standalone application is character based). This poses a serious dilemma since it limits the options of the design of the program. Clearly, a software application that is developed today has to be on par in appearance and quality of presentation with other comparable applications. The application that can be developed using the NEXPERT OBJECT user interface development tool however makes the final product appear outdated and unprofessional. As a result, the typical user would be hard pressed to develop a high level of confidence in the quality of the developed application.

NEXPERT OBJECT does not permit the developer of an application to assign probability values to the successful or unsuccessful evaluation of rules or events in the inference process. Many times there may exist multiple reasoning paths in a knowledge base that lead to the same conclusion. It is not easy for the application developer to assign a value system to these different paths if probability values can not be associated with rules. Probabilities can be assigned to rules indirectly by assigning string values on the right hand side of a rule which then have to be converted later to a numerical value on the left hand side of some other rule. However, this makes the programming of an application difficult.

One of the major shortcomings of the NEXPERT OBJECT program is the fact that the program possesses no inherent file handling utilities. In addition, NEXPERT OBJECT does not support any functions that allow the application developer to define any formatted read and write statements other than the ones included in the NEXPERT flat-file format. NEXPERT OBJECT is not able to create or delete, open or close, and read or write any formatted ASCII files. NEXPERT OBJECT can read and write information only from and to specific proprietary database files and\or its own proprietary database flat-file format. The discovery of this limitation influenced to a large extent the decision to use the NEXPERT OBJECT development shell in a much smaller role, since the limitation would jeopardize the successful development of the Limit States Analysis Module.

NEXPERT OBJECT does not allow the developer to use any higher order decision statements within the rule system such as "IF-THEN-ELSE" or "IF-THEN-ELSEIF-ELSE". The developer can create rules that will behave in the same manner, but a number of rules with related hypotheses have to be programmed to achieve the same effect. The disadvantage is that the rule base becomes more complicated to program as more rules have to be developed to achieve the same results.

NEXPERT OBJECT does not allow the developer to assign a numerical value to an object or property in the right hand side of a rule (i.e. the action side of the rule). If it is necessary to assign a numerical value to a slot, the developer has to assign a string to the object or property which then has to be converted in some other rule to a numerical value. In general it has been determined, that many of the mathematical or logical functions that can be used on the left hand side of the rule can not be used on the right hand side of the rule. As a result the developer has to create more rules than seem necessary to achieve a comparable effect, which complicates the development of an application.

NEXPERT OBJECT does not support any function that would allow the developer of an application to program something comparable to a traditional DO-LOOP. A DO-LOOP can be programmed using NEXPERT OBJECT indirectly, but it is complicated to program and hard to control in the inference process. A DO-LOOP is one of the most versatile programming options since it utilizes a computer in its intended function (i.e. performing repetitive tasks).

NEXPERT OBJECT does not support any graphic functions, static, dynamic, or interactive, that allow the developer to display results derived from the inference process visually to the user of the application. It is therefore impossible for the developer of the application to display any X-Y graphs, line graphs, bar charts, pie charts, or any other common presentation graphics.

NEXPERT OBJECT has a limitation on the length of the character string that can be assigned to a single slot value which is not documented in any of the user manuals. The maximum length of a string that can be assigned to a single slot value is 256 characters. The maximum length of all strings that can be contained in the properties of an object is 1024 characters. These restrictions severely limit the application developer's options.

Neuron Data, the developer and distributor of the NEXPERT OBJECT program, claims that the rule and object structure of it's program can represent fuzzy logic since the inference engine is able to accept an "unknown" or "notknown" value for any of the slot values and still reach a conclusion. Clearly, this is not true. It is true that a slot's value can be defined as "unknown" or "notknown". However, NEXPERT OBJECT's inference engine is not able to reach a conclusion or evaluate a hypothesis until all of the instances leading to the conclusion or hypothesis are determined to be true or false. This indicates, that it is essentially the developer's responsibility to program the application in such fashion, that eventually every slot will have a value that is either true or false (i.e. every slot set to "unknown" or "notknown" has to have a different set of rules associated with it which will eventually change the value to true or false).

Once the limitations of the NEXPERT OBJECT program were assessed, it became obvious that the role of NEXPERT OBJECT had to be redefined in order to develop the Limit States Analysis Module. It was decided, that the application developed using NEXPERT OBJECT had to be confined to becoming just one component of the overall program. Due to the limitations of NEXPERT OBJECT the component would also not be allowed to exert any control on any of the other components. In addition, large proportions of the expertise solicited from the experts were distributed to other components of the program. Some of the solicited expertise was reassigned to be represented in other program components such as the graphic modules, which will be described in detail in the following sections.

The application developed using NEXPERT OBJECT would therefore be used to perform the activities it excelled at, namely to contain the solicited knowledge base that could not be integrated with any of the other components of the Limit States Analysis Module. The applications developed using NEXPERT OBJECT would therefore in its final configuration act as an independent advisor to the user prior to and upon completion of a nonlinear analysis. A more detailed description of the evaluation applications developed using NEXPERT OBJECT 2.0, the interaction of the applications with the other components, and the knowledge contained in the applications is presented in Chapter IV.

DEVELOPMENT OF THE LIMIT STATES ANALYSIS MODULE COMPONENTS

A number of problems, described previously, arose during the development phase of the Limit States Analysis Module prototype which necessitated the redesign of the program. The new Limit States Analysis Module was required to be flexible enough to combine all of the program modules in such fashion that they would be able to operate together and at an optimum level. One of the problems that had to be resolved consisted of the fact that different program modules required different operating environments to perform at an optimum level. The two most significant conflicts are:

- The FORTRAN 77 compiler that the LIMIT B program uses has a proprietary memory management driver called DBOS which allows the program to use expanded and extended memory. Using expanded and extended memory allows any FORTRAN 77 program to execute at a much faster rate on any IBM PC computer. Performance increases are by a factor of two to three. Early on it was deemed necessary to maximize the execution speed of the LIMIT B program because it is calculation intensive. However, the DBOS memory manager conflicted with the MS WINDOWS memory manager (i.e. the IBM PC system does not support the use of more than one memory manager at any one time).

- If a NEXPERT OBJECT application is run as a standalone runtime executable it is run in a character based screen mode. At this point, using the MS DOS environment, it is not possible to run any program module that displays graphics on the screen because the NEXPERT OBJECT inference engine has control over the system. It was determined that these two components needed to be kept completely separate in order to incorporate graphics into the Limit States Analysis Module.

Originally, it was planned that the application developed using NEXPERT OBJECT would have control of all functions during the Limit States Analysis Module program execution. Because of the problems and shortcomings of the NEXPERT OBJECT program that were discovered during the development of the Limit States Analysis Module, the design was redefined completely. It was decided to utilize a system divided into individual components where each of the program modules would be able to operate in its optimum operating environment (i.e. graphic modules use a graphic based screen mode, the NEXPERT OBJECT application uses a character based screen mode, etc.). There are a total of sixteen completely separate program modules that make up the Limit States Analysis Module. A functional description of the individual components of the Limit States Analysis Module is shown in Figure 16 and a detailed description of each of the components is provided in Appendix C.

A number of advantages and disadvantages have been discovered during the development of the individual modules of the Limit States Analysis Module. Since each of the components were completely self-contained, it proved difficult to maintain control over the progress of the analysis. A control program had to be devised that was able to keep track of the state of the nonlinear analysis at any point. In addition, the control program had to be small enough in its memory requirements so it would not impair the execution of any of the other program modules. A batch file approach was used in combination with a very small control program written in MS PROBASIC 7.1 which eliminated any memory allocation problems.

The control program essentially reads the contents of the batch file to determine what module to call. The control program then proceeds to call a batch file that invokes the required operating environment and calls the appropriate executable program module. Once the module has executed, the module will change the contents of the batch file and return control to the control program. Using this method guarantees that control of the state of the nonlinear analysis with the Limit States Analysis Module is maintained continuously throughout the analysis.

A major benefit was realized using the modular design of the Limit States Analysis Module. It became possible for each of the modules to be maintained or upgraded independently without influencing any of the other components.

The modular design developed for the Limit States Analysis Module made it possible to use the standalone expert evaluation applications developed using NEXPERT OBJECT in a protected environment (i.e. an environment in which the user is not able to accidentally change the knowledge base). This allowed the standalone expert evaluations to be used together with the Limit (1) program using the proprietary DBOS memory manager and in combination with the various other program modules that use a graphics based screen mode. In addition, it was now possible to develop a user friendly program interface with graphics based on-line help screens, on-line user and modeling manual, and a professional and timely appearance.

MODIFICATION OF CURVEPLOT

Within the early stages of the knowledge base acquisition it was determined that the existing set of post-buckling member performance model curves was inadequate to model the behavior of most members typically used in the design of a transmission tower. The original set of model performance curves contained a total of eleven curves. The main disadvantage of the existing set consisted of the fact that they were based on experimental tests performed on long, slender compression members. It was therefore difficult to select an appropriate post-buckling member performance curve for any of the short or intermediate members in a structural model. Another disadvantage was that each member performance model curve had to be selected using the original version of the CURVEPLOT program which is cumbersome to use, because it was developed in a main frame computer environment.

A number of solutions were devised to eliminate these problems. First, it was decided to develop a new improved CURVEPLOT program using MS PROBASIC 7.1 that would permit the user of the Limit States Analysis Module to select the model performance curves more efficiently. The new CURVEPLOT program module would perform most of the tasks associated with the selection process for the user. In addition, the new CURVEPLOT program would allow the user to select model curves interactively from within the Limit States Analysis Module Graphic module for any of the members in the structural model, rather than for one member at a time.

Second, a new set of post-buckling member performance curves was developed based on information obtained from physical member tests described in the following section. The new set of curves was required to be appropriate to model the post-buckling member behavior of single equal and unequal leg angles and double angles. The new set of curves that was developed contains a total of thirty curves and is fundamentally different from the old set of model curves. The new set of curves consists of five groups of curves with similar characteristics. Each of the curves is normalized and fully adjustable to appropriately model the post-buckling performance of any member (i.e. the shape and other characteristics of each curve can be adjusted based on member strength and geometric properties). The new set of model curves is shown in Figures 17 through 21. The model curves with the lower designation numbers represent the behavior of short, stubby compression members. The model curves bearing the higher designation numbers represent the behavior of long, slender compression members.

Finally, a test program was developed to determine the appropriateness of the new set of post-buckling member performance model curves. A number of existing transmission tower designs were selected at random to verify the new set of model curves. For each of the members of the randomly selected towers post-buckling model curves were assigned in an attempt to determine how well the set of model curves represented the post-buckling behavior of the structural members. Upon completion and evaluation of the test it was determined that the new set of model curves produced excellent correlation with the experimental post-buckling member performance curves. The new set of model curves was then integrated into the new CURVEPLOT and LIMIT B program.

EXPANSION OF THE POST-BUCKLING MEMBER PERFORMANCE DATABASE

A limit states finite element analysis is very much dependent upon the selection of the proper post-buckling member performance. If an inappropriate member performance is specified LIMIT B will predict an inaccurate collapse load factor for the structural system analyzed.

Since LIMIT B was specifically developed for the analysis of transmission towers, the experimentally obtained post-buckling member performances contained in the original CURVEPLOT program are mostly for single, equal leg, long, slender steel angle members tested in compression (5). The performance of single steel angles in compression is affected by the member's effective slenderness ratio (KL/R), the type of end restraints, the eccentricity of the applied compressive load, and by intermediate connections such as bracing, and lap or butt joints.

In a LIMIT B analysis it is the after failure or post-buckling capacity of the member that of primary interest. A nonlinear LIMIT B analysis is identical to an elastic analysis until any one of the members in the structure reaches its elastic capacity. This was done to extend a classical elastic analysis to include after failure member performance. At this instance, the load carrying capacity of that member will either remain constant or start to decrease. This will change the type of analysis from an elastic to a nonlinear LIMIT B analysis (See Figure 22). At this point, the post-buckling member performance model curve that has been selected for that particular member will become important and decisive input to the LIMIT B program. It is therefore imperative to select the most appropriate post-buckling member performance to represent the geometric and parametric characteristics of the member.

The factor that affects the post-buckling member performance most is the slenderness ratio (KL/R). A distinction is made between "short" members and "long" members. A short member is defined as a member that has a KL/R ratio that is less than or equal to the critical KL/R ratio C_c (20, 21) which is defined as:

$$C_c=((2*E*PI^2)/F_y)^0.5) \quad (7)$$

E - Young's Modulus of Elasticity

F_y - Nominal Yield Stress

C_c= 126 for A36 steel (F_y=36ksi).

A long member is defined as a member that has a KL/R ratio that is larger than the critical KL/R ratio C_c.

One problem encountered during the development of the Limit States Analysis Module was that the post-buckling member performance database contained in the original CURVEPLOT program was based on experimental tests of long, slender single angle compression members only. This was because the original use of the LIMIT B program was in the analysis of tension only tower configurations. A tension only tower is a structural system that resists all of the lateral load applied to the structure through tension braces (i.e. the compression capacity of the second member of the brace is assumed to equal zero). During the early stages of the knowledge acquisition it was determined that it proved to be very difficult to select an appropriate post-buckling member performance for many members since their characteristics differed from the characteristics of the members represented in the database. It was decided to expand the member performance database to contain information about all of the members typically used in a transmission tower. Steel angle members used in a transmission tower are single angles with equal leg sizes and a slenderness ratio in the range of 20 to 350, single angles with unequal leg sizes and a slenderness ratio in the range of 20 to 350, and double angles with equal or unequal leg sizes and a slenderness ratio in the range of 20 to 250. In addition, most towers contain connections such as lap or butt joints which are usually located somewhere along the length of the built up member. A test program was developed that would supplement the information already contained in the CURVEPLOT post-buckling member performance database.

The test program that was developed consisted of 31 compression tests of single steel angles with equal legs, 44 compression tests of single steel angles with unequal legs, 28 compression tests of double angles with equal legs, and 44 compression tests of double angles with unequal legs. In addition, a number of tests would be performed to assess the effects of lap or butt joints on the post-buckling behavior of steel angles.

Steel angles come in a multitude of different leg sizes and leg thickness combinations. A validation process had to be devised to determine the most appropriate selection of typically used steel members in an attempt to minimize the extent of the test program. It was decided, that it would be beneficial to the test program if the tests were performed for member sizes most frequently used in the design of transmission towers. Existing tower designs were analyzed in order to develop a histogram of the member sizes typically used in the design. The member sizes that were selected for the test program were then determined from the histogram for both single and double steel angles. The selection of the member sizes and member configuration for the lap and butt joint tests were developed in much the same manner.

Next, a member test setup and configuration was developed that would allow the results to be integrated with results previously obtained (5). It was decided to develop a computer acquisition system, hardware and software, that would allow the results of the member tests to be stored on magnetic media. It was anticipated, that storing the test results on magnetic media would reduce errors in the information gathering process, make it easier to process the data for whatever intended purpose, and allow the data to be stored in an accessible manner for future use.

Upon completion of the member tests the results obtained were verified in a number of different ways. First, the experimentally obtained ultimate load capacities of the members were compared to theoretical values computed using the current design standard (21) in the transmission tower industry. Second, the experimentally obtained ultimate load capacities of the members were compared to the results obtained using a finite difference analysis program. The experimental results obtained from the tests and the results of these comparisons have been reported by Bathon (26) for single steel angles and by Huntley (28) for double steel angles.

Significant differences were found between the experimentally obtained and theoretically computed ultimate capacities of the members (26,28). It was assumed, that the differences were caused by the normal framing eccentricities as they are defined in the current design standard (21). In order to verify this assumption it was necessary to test a number of steel angles concentrically.

In addition, a small test program was initiated to assess the influence of load eccentricities on the ultimate capacity of steel members and their post-buckling behavior. The capacity values obtained from the concentric load tests compared perfectly to the theoretically computed capacity values using the current design standard (21) and thus verified the results obtained by the test program. A detailed discussion of the verification process has been presented by Bathon (26) and Huntley (28).

It has been determined that the post-buckling performance of a short member with a KL/R of 60 will show a rapid unloading after the ultimate capacity of the member has been reached. Since this behavior is typical of constitutive relationships of brittle materials such as concrete or ceramics it is called a "brittle" failure despite the fact that steel is a rather ductile material. A long member with a KL/R of 180 will show a constant force plateau for its post-buckling performance that can be defined as a "ductile" failure. The experiments have shown that there is a smooth transition from the brittle to the ductile failure mode, as can be seen in Figures 23 through 25 which are results obtained experimentally from physical member tests (26) for KL/R values of 60, 120, and 210 (i.e. a short, intermediate, and long member) respectively.

Last, the experimental results of these tests were used to develop a new expanded post-buckling member performance data base for the CURVEPLOT program that would predict the nonlinear behavior for any single member with equal or unequal legs, and double angle member with long legs back to back. New regression data base models were created from the experimental test results that were integrated into the modified CURVEPLOT program. A detailed description of the development process is documented by Robinson (27). A three-dimensional representation of the newly developed regression models is shown in Figures 26 through 28. The predicted nonlinear member performances as they are predicted by the appropriate regression models are used as input to the LIMIT B finite element program and therefore the Limit States Analysis Module.

DEVELOPMENT OF THE TOWERPLOT GRAPHICS MODULE

One of the significant reasons in favor of the development of the Limit States Analysis Module consisted of the fact that the user of the LIMIT B program had no visual aids available during the nonlinear analysis. It was felt that it was absolutely essential to the successful development of the Limit States Analysis Module to supply such visual aids.

A need assessment study was undertaken that determined what visual aids would prove beneficial to the typical user of the Limit States Analysis Module. The study determined that it would be beneficial to the user to have the following visual aids:

- Graphical display of the structural model prior to and upon completion of the analysis.

- Graphical display of all of the applied dead and live loads prior to and upon completion of the analysis.

- Graphical display of the member and joint labels.

- Graphical display of inelastic and \or critical members upon completion of the analysis.

- Graphical display of original and deflected shape of the structural model upon completion of the analysis.

- Graphical display of all the applied artificial restraints and specified deflections prior to and upon completion of the analysis.

It was decided to develop the Towerplot graphics module using the MS Probasic 7.1 programming language which would provide the above mentioned options to the user. Problems that were encountered during the development of the Towerplot graphics module were restricted to debugging of the source code and memory size limitations of the variable arrays. Upon completion of the program development the graphics module was integrated with the Limit States Analysis Module. A detailed description of the Towerplot graphics module and all of its options is presented in Appendix C.

DEVELOPMENT OF THE LOADHISTORY GRAPHICS MODULE

One of the unique features of performing a nonlinear analysis using the LIMIT B or ES program is the load-history log that the program maintains if the correct flag is set in the control parameters. The load-history log assists the user in determining the collapse load mechanism of the structure. Originally, the log was kept in terms of load and deflection values of each member for each particular load factor. This requires the user to manually plot each individual pair of values for each load factor in order to determine the break point (i.e. the point at which the member starts to perform inelastic). Many members usually have to be plotted until the collapse mechanism most likely to occur can be determined.

It was decided to develop a graphics routine that would assist the user in the interpretation of the load history log and therefore the determination of the collapse load mechanism. The final graphics module would have to be able to plot the loadhistory of a number of members simultaneously. One of the problems encountered in the development of the Loadhistory graphics module was that the load history log file is usually rather large. A memory swapping routine had to be devised in order to avoid variable array memory limitation problems. A detailed description of the Loadhistory graphics module and all of the options that are available to the user can be found in Appendix C.

DEVELOPMENT OF THE NEW LIMIT STATES ANALYSIS ALGORITHM

Originally, a nonlinear analysis using the LIMIT B program was performed using the algorithm described in the following paragraphs. Algorithm is defined within this section as a rule or procedure that is followed to solve a problem. The original solution algorithm consisted of running the LIMIT B analysis assuming elastic member performance behavior first, and then to run the program again assuming bilinear member behavior. After multiple iterations of assigning bilinear curves to the overstressed members and rerunning the analysis the solution would eventually stabilize. At this point the control parameters were changed and the process of iteration would start all over. This process of assigning curves, running the analysis, and redefining the control parameters would continue many times until eventually the solution stabilized, or ridiculous results were produced.

Next, the same procedure as discussed above would be repeated for the LIMIT B analysis using normalized post-buckling member performance model curves. The main difference at this point was that every time a member behaved inelastically the user was required to specify an appropriate normalized member performance curve number for the member. Each normalized curve had to be denormalized based on the strength and geometric characteristics of the member. The denormalization process had to be done manually by the user multiple times for each member individually until an appropriate match between the predicted and the model curve was found. Both, the selection of the control parameters and the selection of the post-buckling member performance model curves had to be reiterated until the solution would eventually stabilize and a justifiable collapse load factor was computed. A typical nonlinear analysis using the old LIMIT B solution algorithm took an average of two person-weeks.

A new improved solution algorithm was developed for the nonlinear analysis using the modified LIMIT ES program. The new algorithm requires the user to perform a checkrun with the specified input file in order to verify the structural model. As in the case of the old algorithm, the user then proceeds to perform an elastic analysis which is used to verify the structural model further by comparing the results to results obtained by other analysis programs. This is not a profound change, but is required in the new algorithm as a result of much consultation with the human experts.

Next, the user performs a nonlinear analysis with the LIMIT ES program using bilinear member performance behavior. Every member that becomes overstressed during the analysis is automatically assigned a bilinear member performance curve until all of the critical members are defined and a valid collapse load factor is calculated. The expanded set of control parameters allows the user to obtain a valid collapse load factor within one bilinear analysis.

Last, the user performs a nonlinear analysis with the new LIMIT ES program using normalized post-buckling member performance model curves. Normalized performance curves can now be assigned to each of the members previously determined to be or to become critical. The curves are assigned interactively by the user using the newly developed CURVEPLOT graphics module. The graphics module denormalizes the normalized member performance curves for the user and displays it on the screen together with the predicted performance curve obtained from the experimental database. In addition, the graphics module displays to the user the displacement value at which the member is performing. This allows the user to find the most appropriate model curve in the first iteration. Once the user has selected a normalized curve for each of the relevant members, the analysis is run and a collapse load factor is calculated. If the proper control parameters are selected the analysis usually does not have to be rerun and the collapse load factor calculated is likely to be correct. A typical nonlinear analysis using the new LIMIT ES program can now be performed within 3 to 5 hours.

The new algorithm is more advantageous since the built-in controls protect the analysis from runaway solutions and at the same time allow the program to focus directly on a realistic solution. In order to implement the new algorithm it was necessary to add a number of control parameters to the program. These control parameters essentially indicate to the program when and how a bilinear curve number should be automatically assigned during the analysis using bilinear and also normalized member performance behavior. In addition, the control parameters trap any overstressed elastic members and automatically terminate the analysis if this phenomena occurs for maximum computational efficiency.

The new nonlinear analysis algorithm was also changed to keep track of trouble members. Trouble members are members which cause the solution process to terminate since the program is not able to determine a solution compatible with the after failure member performance within the convergence criteria range specified. It is these trouble members that are most likely to promote collapse of the structural system in combination with other members involved in the mechanism.

MODIFICATION OF THE LIMIT ANALYSIS PROGRAM

During the development of the Limit States Analysis Module a number of modifications had to be made to the original LIMIT B program. It was decided to upgrade the LIMIT B program to adopt it to today's state of the art technology (i.e. to change the program in such manner that it can take advantage of today's hardware and software technology) and to facilitate integration of the program into the Limit States Analysis Module.

A number of changes were made to the original LIMIT B program and they are described below:

- The LIMIT B program was moved from the VAX\VMS environment to the MS DOS environment. The program was recompiled using the SALFORD FORTRAN 77 compiler which allows the program to utilize expanded and extended memory.

- The input and output file structure of the LIMIT B program was changed. The program has now the ability to produce a checkrun output file, a full analysis output file, a load history file, and a runtime execution log file.

- Control parameters were added to the LIMIT program which allow the program to perform a checkrun and allow a faster program execution.

- Extensive error trapping routines were included in the program that protect the analysis from unexpected runtime termination.

- The program was changed to accept the newly developed set of 30 post-buckling member performance model curves.

- The format of the output file created by LIMIT was changed and expanded. More information is now written to the output file which is used by the expert output evaluation and which makes it easier for the user to determine the accuracy of the analysis.

- The program was changed to facilitate the use of the new nonlinear limit states analysis algorithm that was developed and is discussed in the previous section.

The majority of the changes made to the LIMIT B program mentioned above were made to the program to eliminate problems that occurred during the development of the Limit States Analysis Module. For example, it was necessary to allow the user the option of performing a checkrun prior to the analysis. The checkrun option ensures to the user the correctness of the structural model and allows the user to view the model using the Towerplot graphics module prior to a full analysis. Extensive error trapping routines were incorporated into the program to protect the user from having the program terminate before it can calculate a collapse load factor. The error trapping routines also guarantee that the control is always returned back to the Limit States Analysis Module if the program is terminated unexpectedly. For more detailed information about the changes made to the program consult the LIMIT ES user manual.

DEVELOPMENT OF THE BANDWIDTH OPTIMIZATION MODULE

Preliminary tests of the Limit States Analysis Module showed that the nonlinear analysis using the LIMIT ES program took an unacceptable amount of time to calculate a collapse load factor. It was decided to develop a method that would improve the performance of the nonlinear analysis. A numerical algorithm was derived that would minimize the bandwidth of the structural stiffness matrix to improve the performance of the analysis. The bandwidth of a matrix is the number of non-zero entries in each row of the matrix along the main diagonal. Minimizing the bandwidth of the structural stiffness matrix will allow the program to use a smaller amount of memory allocations and reduces the time necessary to solve the equations.

The Bandwidth Optimization module was written to allow the user to either specify a seed joint manually, or have the algorithm compute the bandwidth for each joint in the model. This was done to give the user the option to gain expertise with the subject of structural analysis using a what-if scenario. Once the seed joint is specified, the algorithm proceeds to renumber all of the joints appropriately until the new stiffness matrix is defined. The algorithm then proceeds to calculate the bandwidth of the resulting stiffness matrix. In the automatic mode, the algorithm will calculate the bandwidth for each of the joints (i.e. evaluate the bandwidth of the matrix for each joint of the model). Upon completion of the analysis the program will report the optimum seed joint that produces the minimum bandwidth of the stiffness matrix.

Once the Bandwidth Optimization module was completed, it was determined that the module itself used an unacceptable amount of computational time. It was determined that most transmission tower structures are symmetric about at least one of the two horizontal axes. The decision was made to use the module in such fashion that it would take advantage of the symmetry condition. The module was therefore changed to check the bandwidth of the matrix for only one of each of the four joints in the main body of the tower and for only one of each of the two joints in the bridge section of the tower. Using this approach results in a more efficient algorithm that produces identical results.

Upon completion of the Bandwidth Optimization program, the module was integrated into the Limit States Analysis Module. Performance tests showed that using the automatic Bandwidth Optimization module reduces the computational time required for a nonlinear analysis using the LIMIT ES program by up to 65 %. A more detailed description of the Bandwidth Optimization module and its options is presented in Appendix C.

REVISION OF THE LIMIT USER MANUAL

Early on in the development phase of the Limit States Analysis Module it was decided to provide the user with the ability to use an on-line help feature. Many changes, as previously discussed, had been made to the original LIMIT B program to develop the new LIMIT ES program. The original LIMIT B documentation therefore had to be changed to reflect these changes.

Once the LIMIT documentation was updated (i.e. figures, text, options, and examples) using a traditional word processor an ASCII based random access file version was developed that could be used by the user from within the Limit States Analysis Module. A random access file format was used to improve the access time to the user manual. One of the problems that arose during the development of the ASCII version consisted of the fact that the figures contained in the document version could not be displayed. It was decided to solve the problem by programming function keys that would activate a picture viewer which displays the bitmap figures. The final updated version of the on-line user manual was then integrated into the Limit States Analysis Module. A more detailed description of the on-line user manual is given in Appendix C.

DEVELOPMENT OF THE LIMIT MODELING MANUAL

The knowledge base solicited from the two experts was used in the development of the pre-processor, the expert input evaluation, and for the expert output evaluation. Once the programming phase of the expert evaluations was completed, it was decided to document the experiences related to the nonlinear analysis using the new LIMIT ES program with post-buckling member performance curves. In particular, it was deemed beneficial to document in detail the process the user undergoes to select appropriate post-buckling member performance model curves. The documentation of the solicited knowledge would preserve the expertise for future uses such as an expansion or update of the Limit

States Analysis Module.

At the time of completion of the LIMIT (23) modeling manual it was decided that it would be beneficial to include the information contained in the modeling manual as an on-line module in the Limit States Analysis Module. Again, as in the case of the LIMIT user manual an ASCII based random access file format was utilized to develop the on-line modeling manual. All of the figures contained in the original manual were recreated in a bitmap format and were linked to keyboard function keys. The on-line LIMIT modeling manual was then integrated into the Limit States Analysis Module.

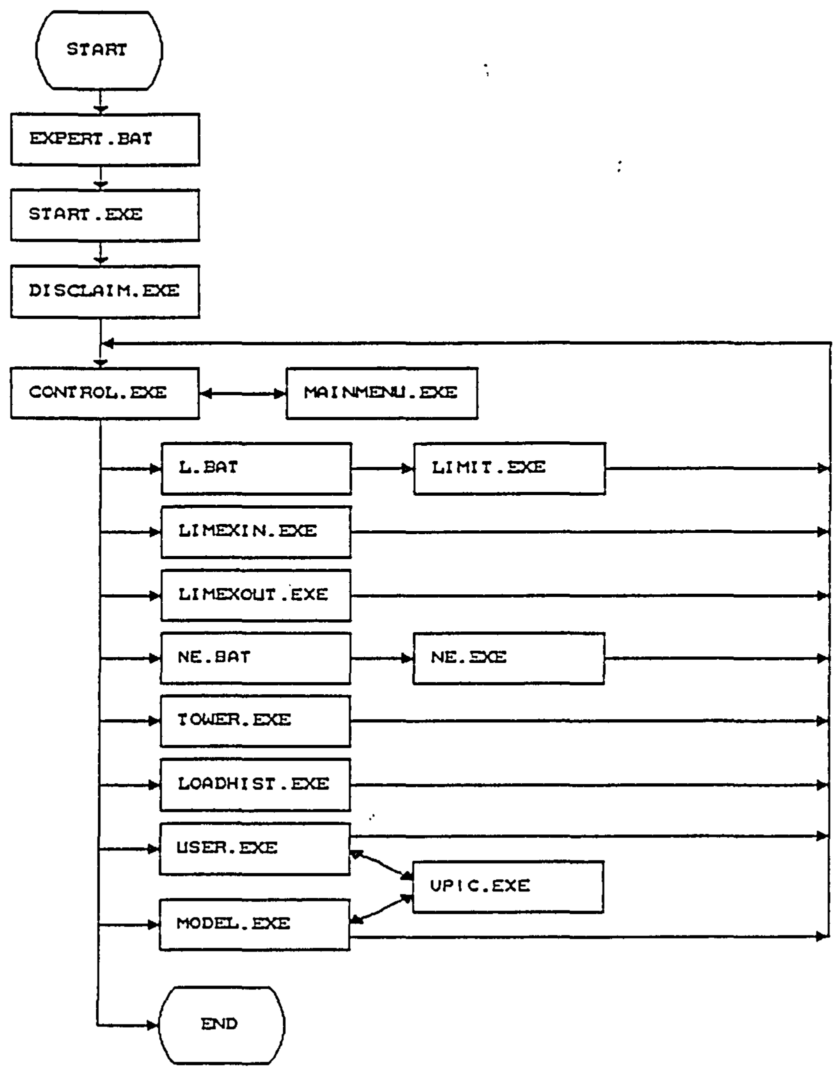


Figure 16. Functional Description of Limit States Analysis Module Program Modules.

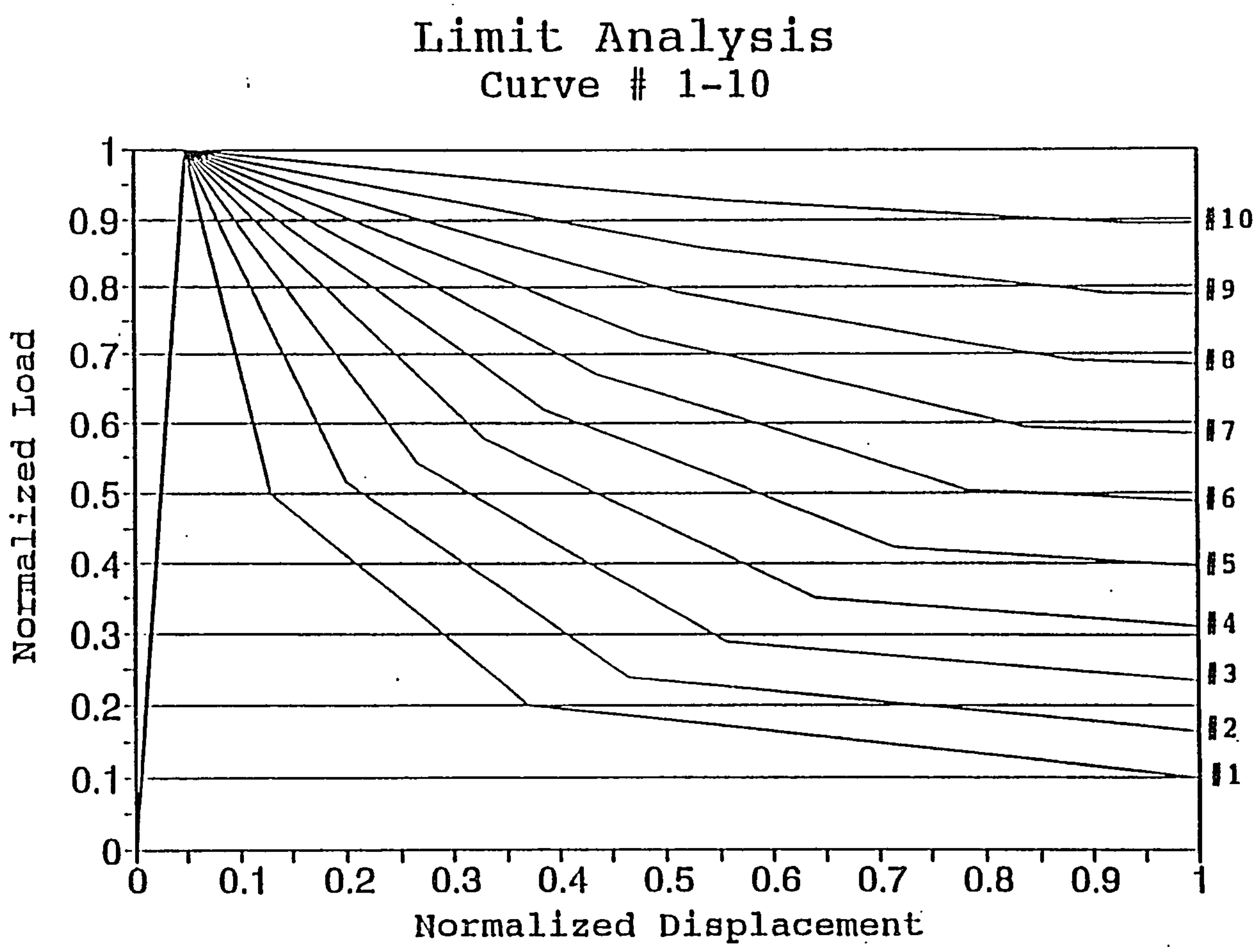


Figure 17. Recommended Normalized Curves 1-10.

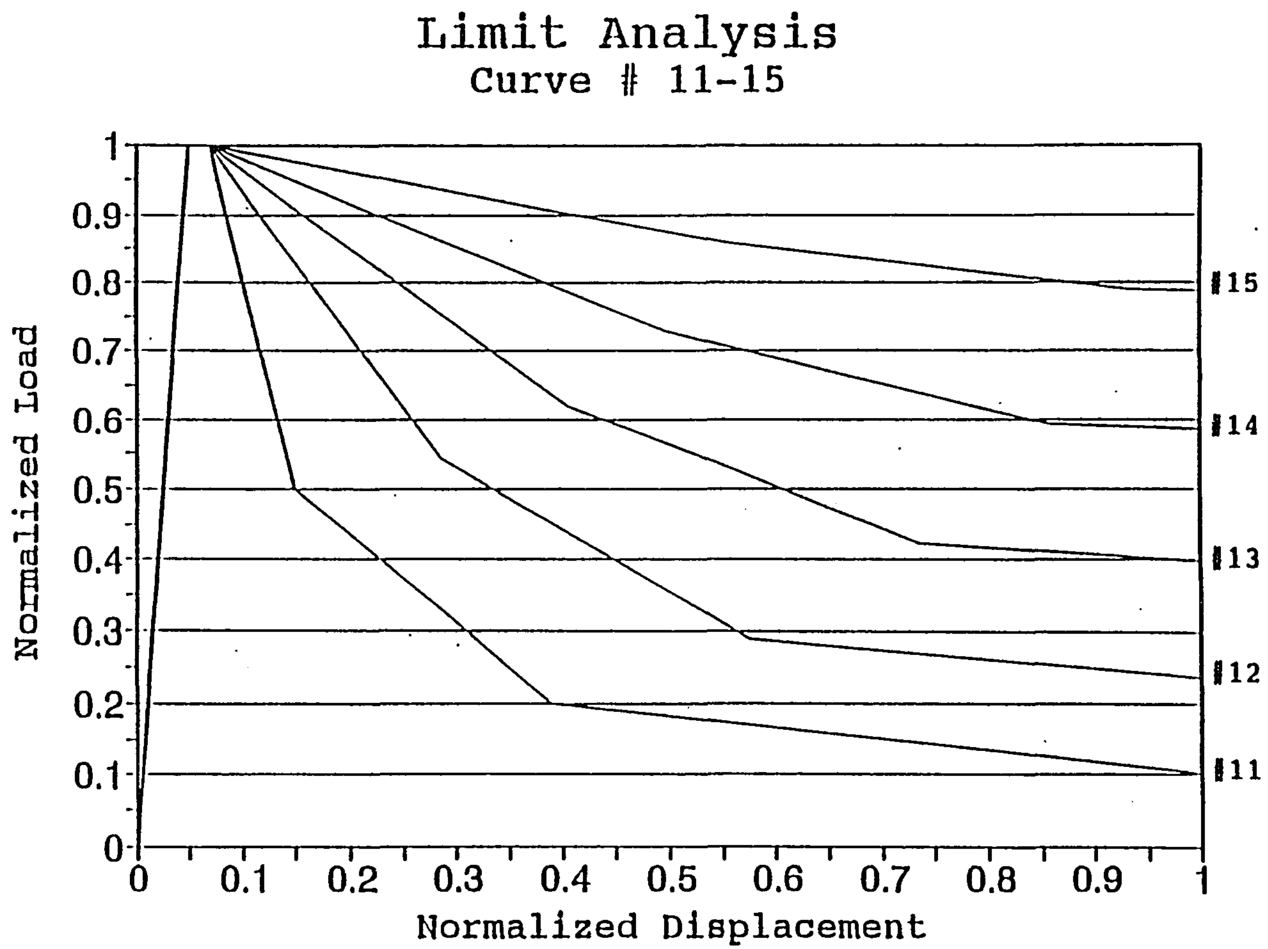


Figure 18. Recommended Normalized Curves 11-15.

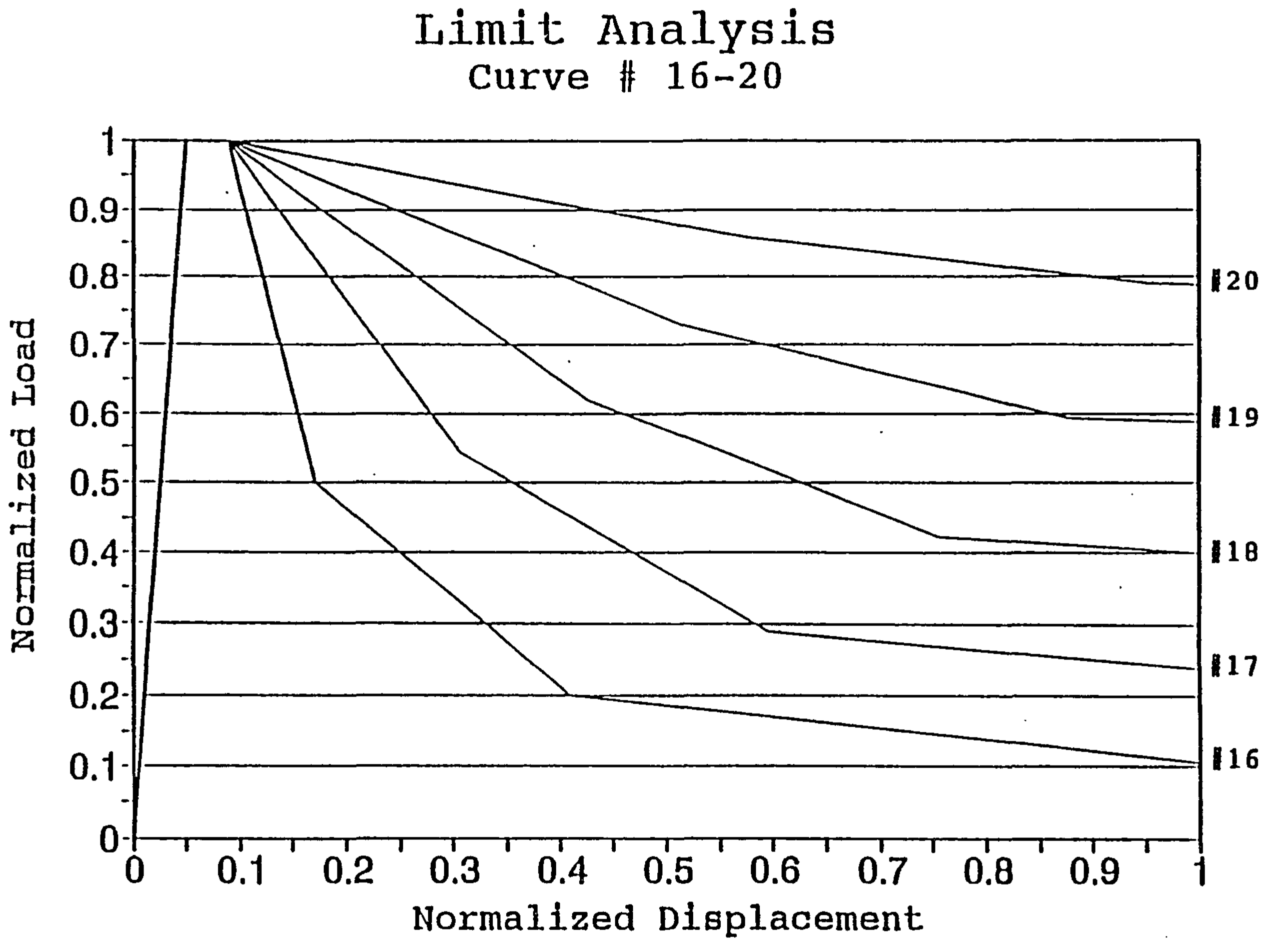


Figure 19. Recommended Normalized Curves 16-20.

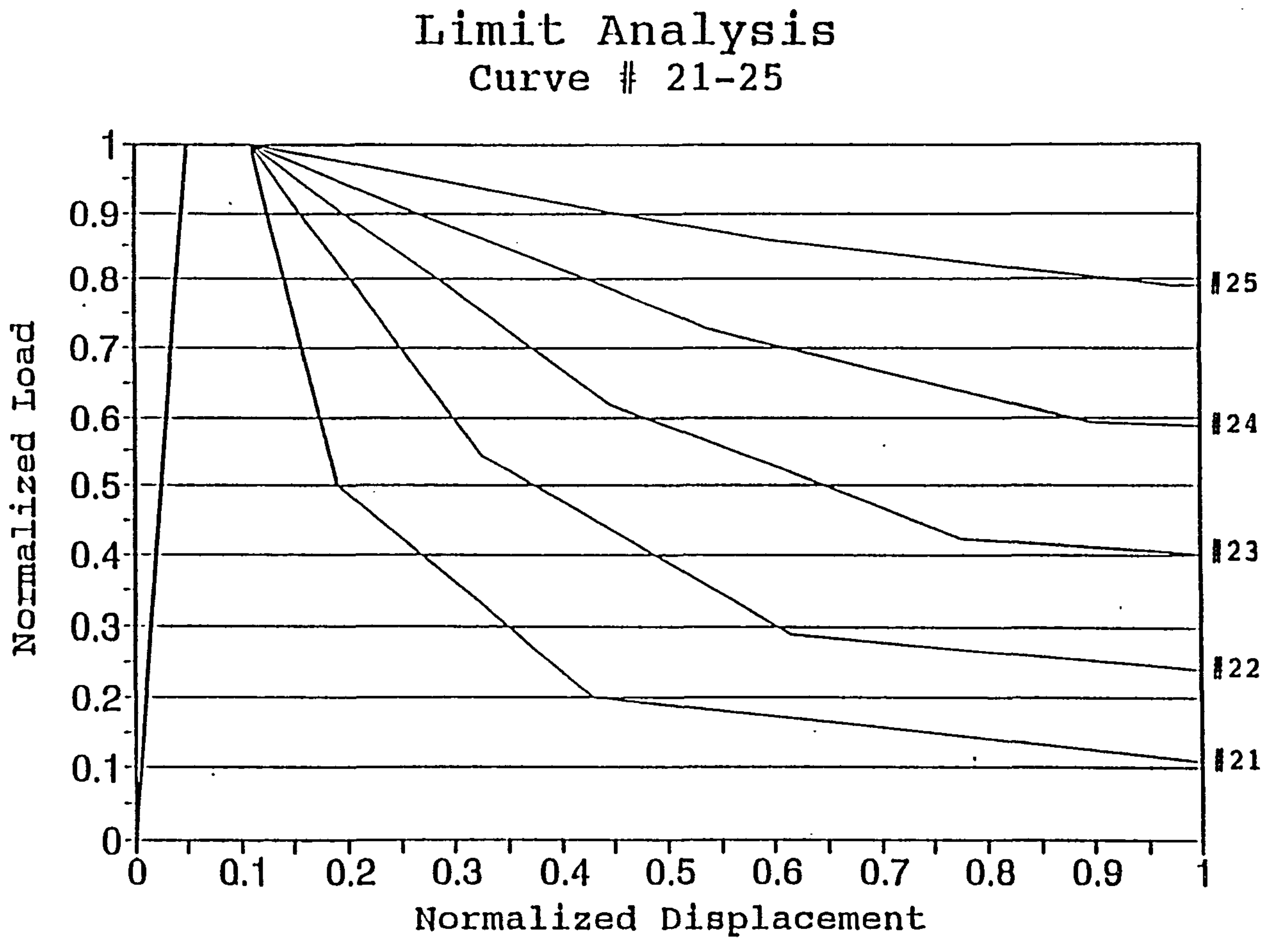


Figure 20. Recommended Normalized Curves 21-25.

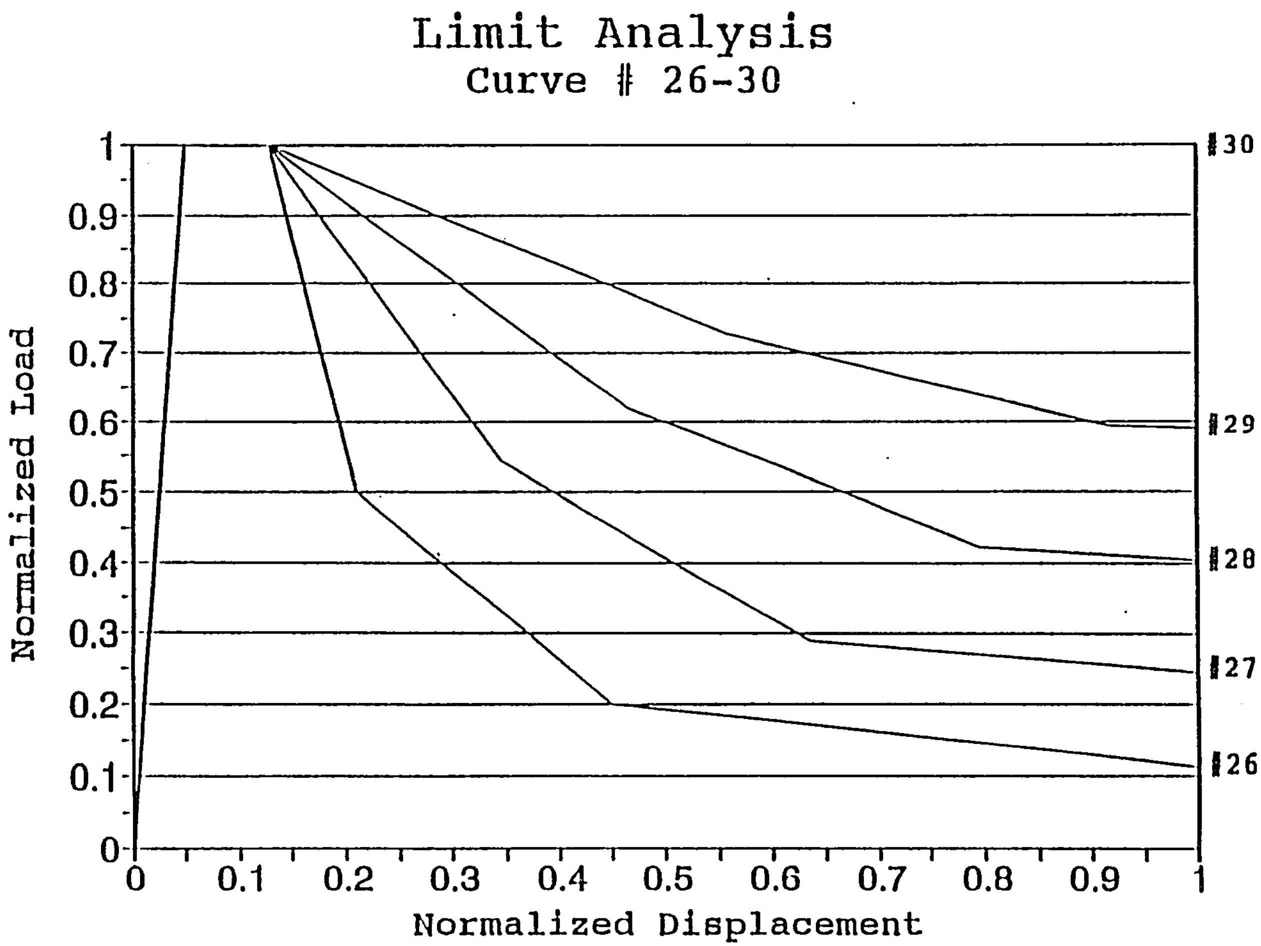


Figure 21. Recommended Normalized Curves 26-30.

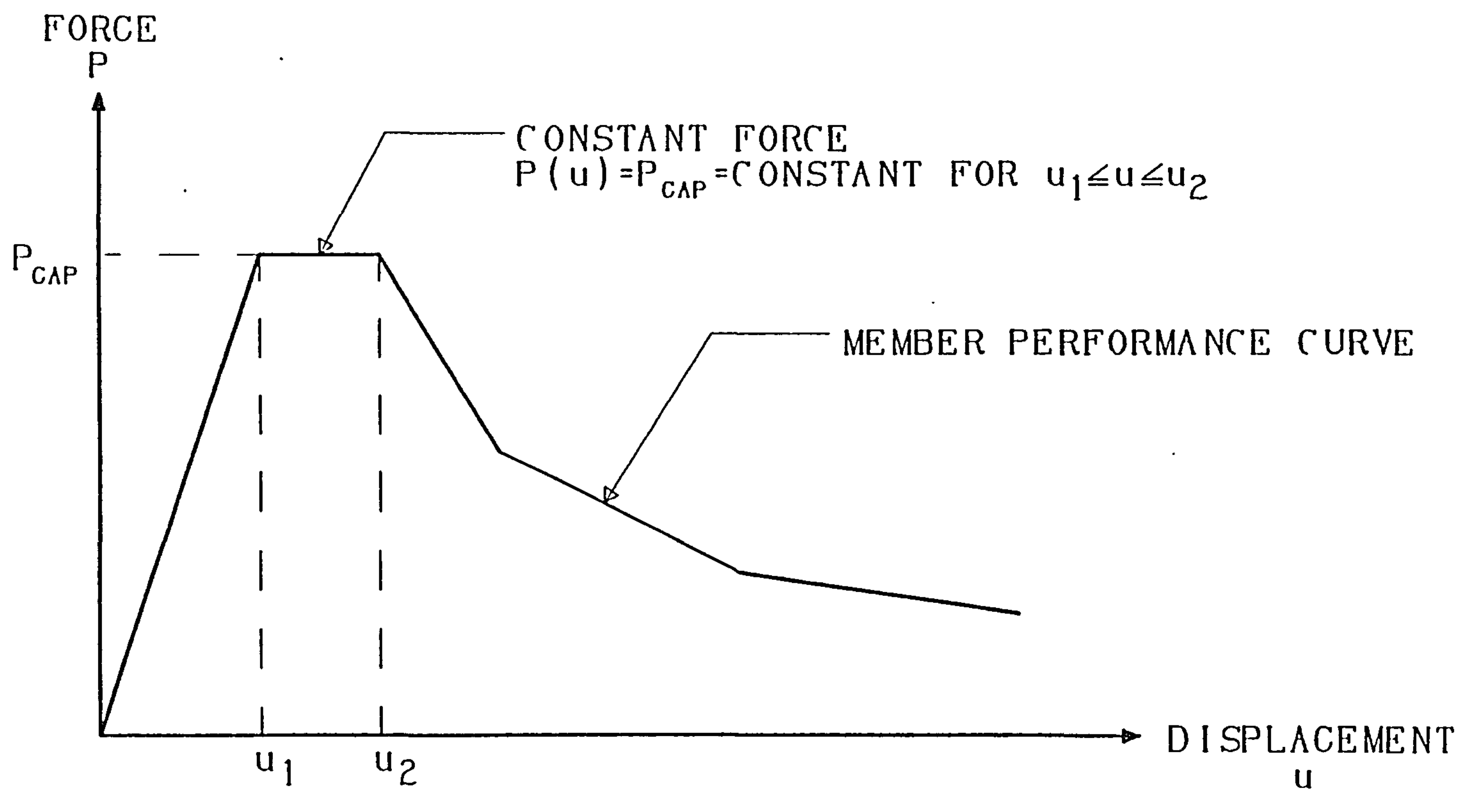


Figure 22. Constant Force Member Performance.

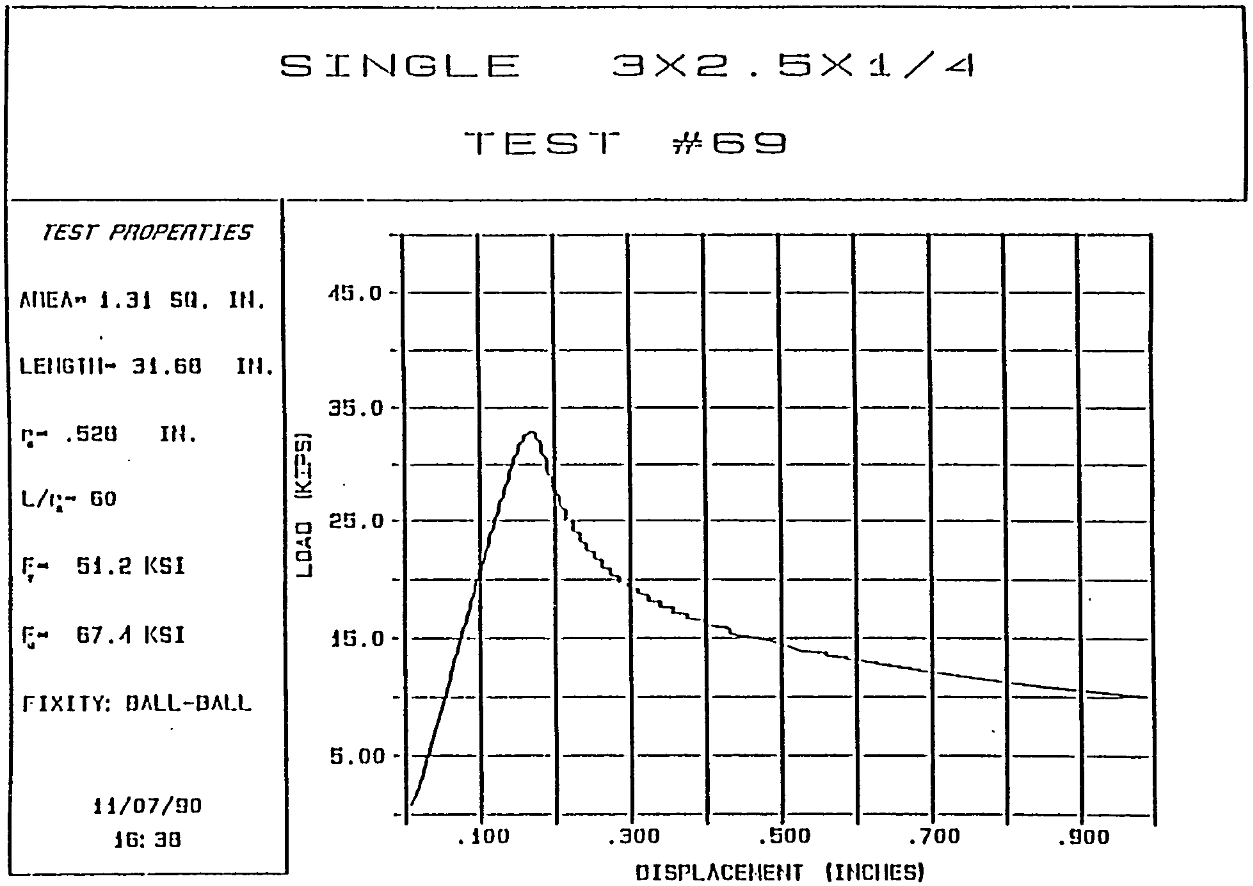


Figure 23. Experimental Test Results - KL/R=60.
(Adapted from Bathon, P.128 (26))

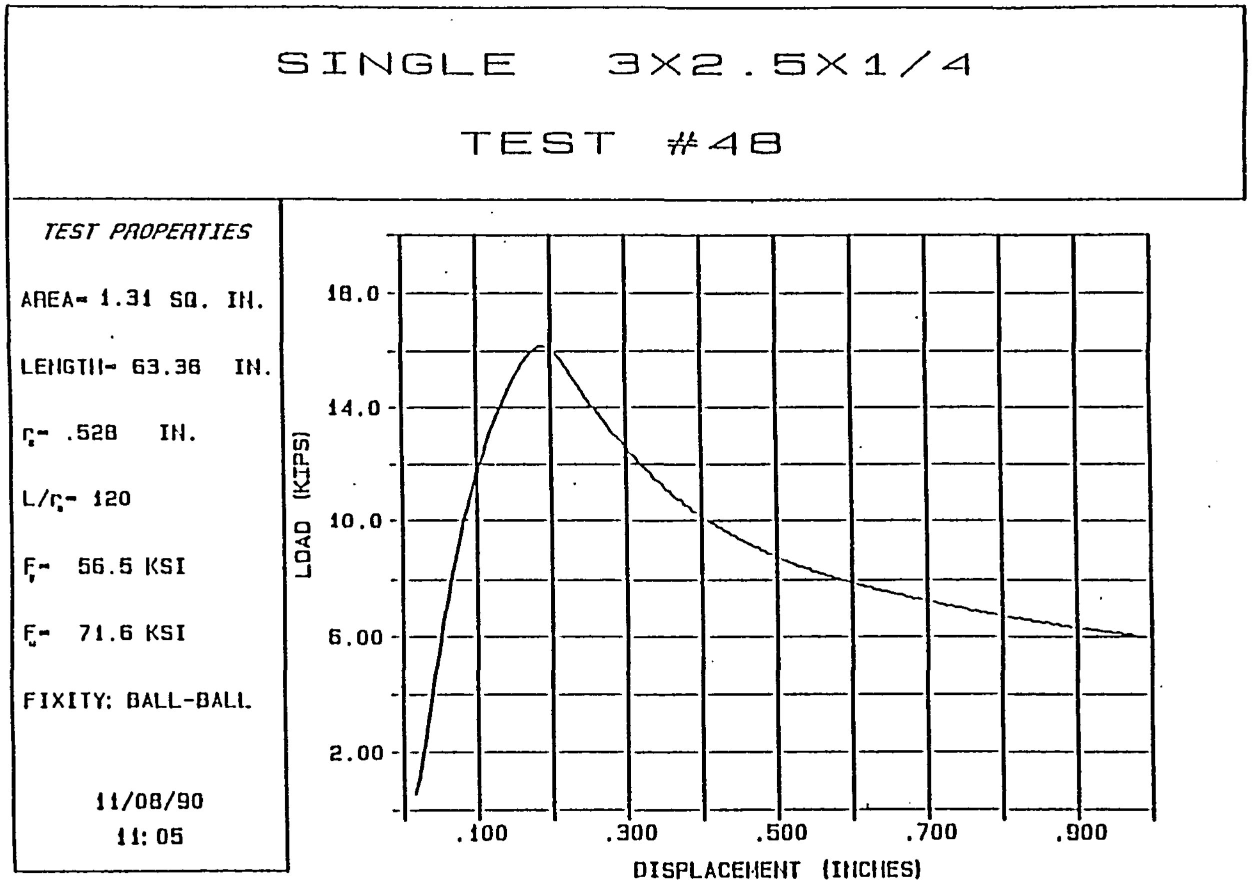


Figure 24. Experimental Test Results - KL/R=120.
(Adapted from Bathon, P.107 (26))

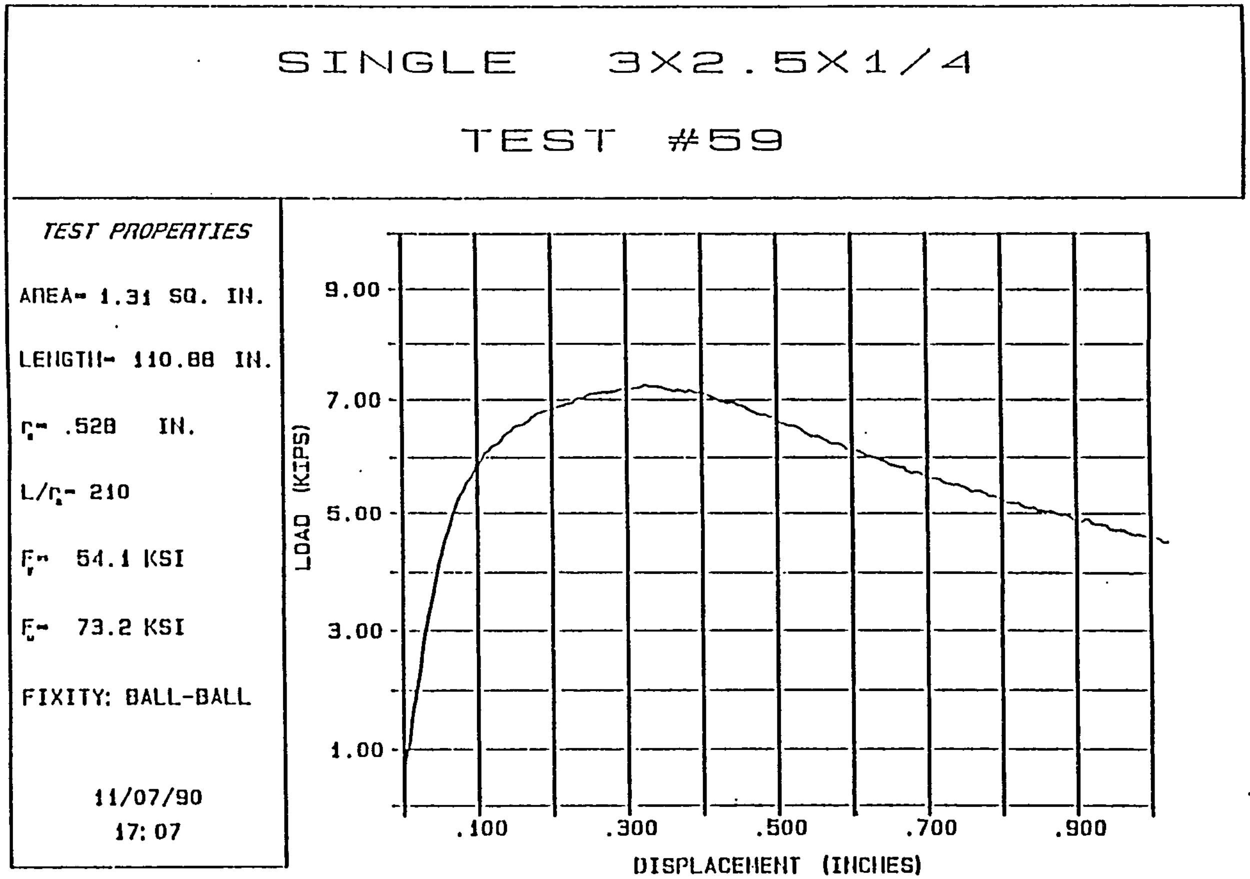


Figure 25. Experimental Test Results - KL/R=210.
(Adapted from Bathon, P.118 (26))

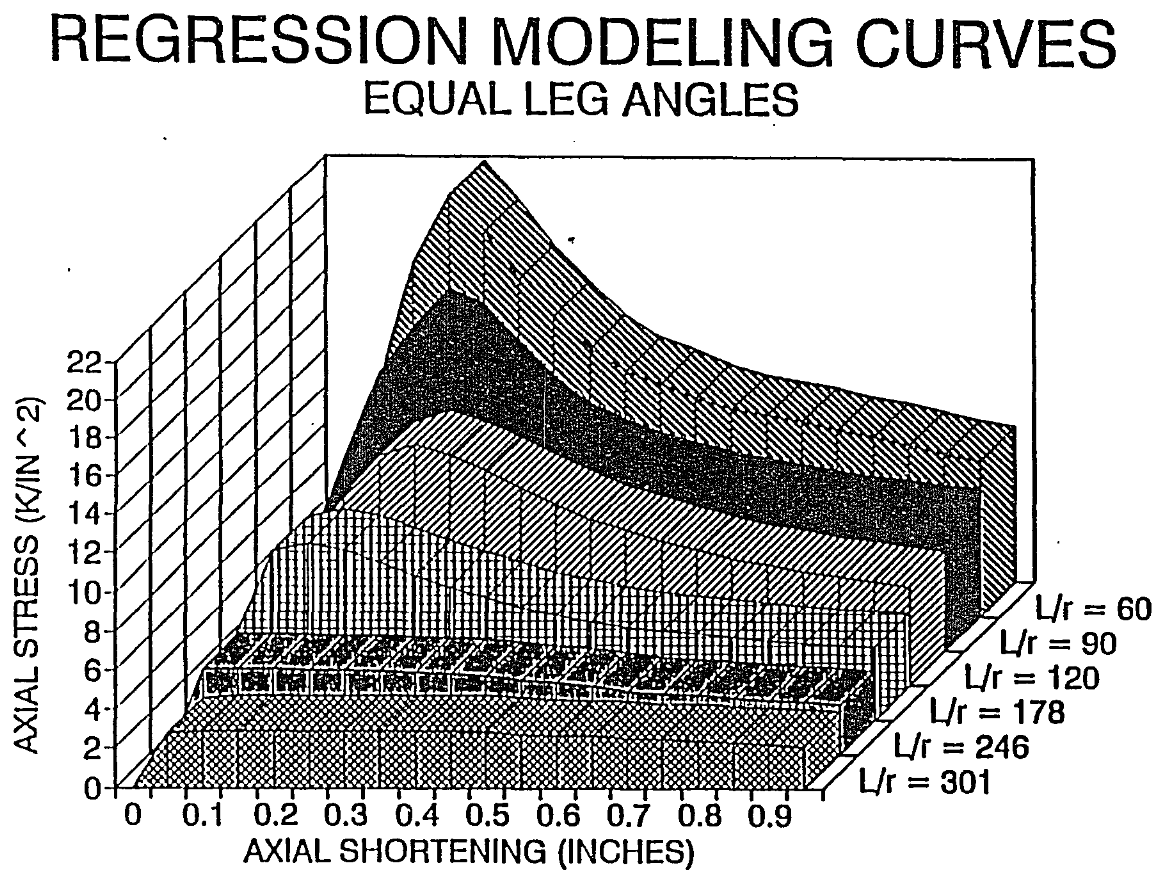


Figure 26. Regression Data Base - Equal Leg Angles. (Adapted from Robinson, P.11 (27))

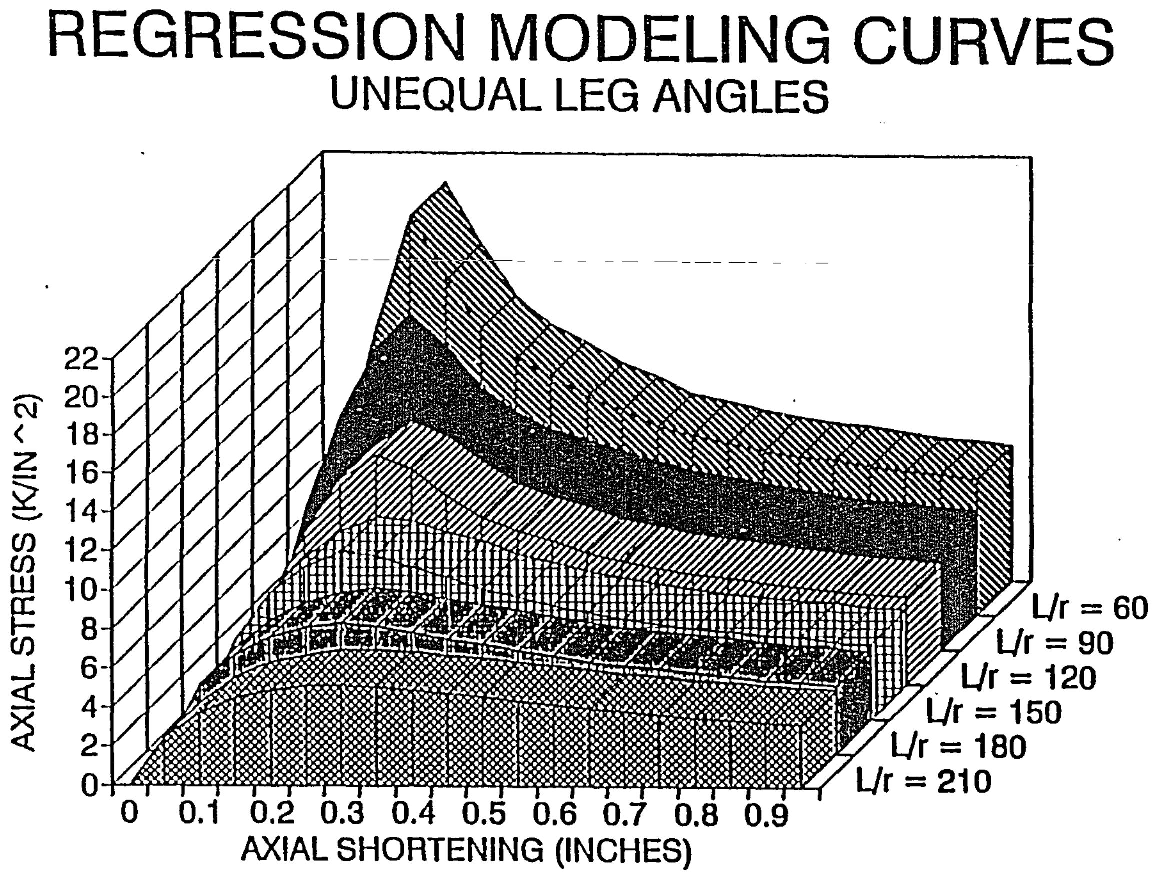


Figure 27. Regression Data Base - Unequal Leg Angles. (Adapted from Robinson, P.12 (27))

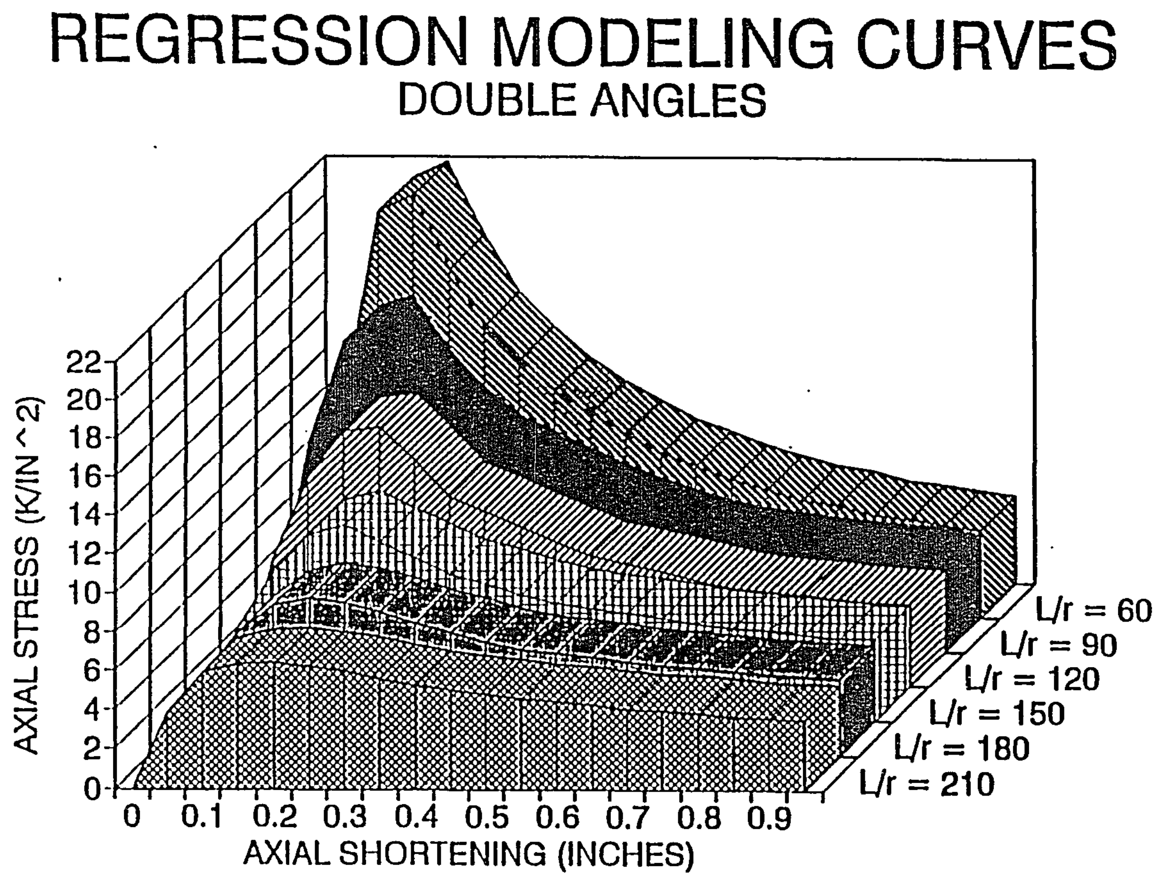


Figure 28. Regression Data Base - Double Angles.
(Adapted from Robinson, P.13 (27))

CHAPTER IV

DEVELOPMENT OF THE EVALUATION MODULES OF THE LIMIT STATES ANALYSIS MODULE

EXTRACTION OF THE KNOWLEDGE

The process of defining the expert system application area and therefore the necessary knowledge (facts, rules, heuristics, and procedures) associated with the application is called the knowledge acquisition. In the knowledge acquisition process, the expert system application developer attempts to extract and define from documents and directly from the human expert(s) the manner in which the problem or situation would usually be resolved. As a result of the knowledge acquisition, the application developer obtains a complete specification of the knowledge to be implemented within the expert system application.

The knowledge acquisition is usually accomplished through interviews of the domain expert(s) by the application developer, in which the application developer attempts to elicit the knowledge from the expert(s) by specific and iterative questioning during interviews.

Hypothetical situations are proposed by the application developer in which the expert is encouraged to discuss in detail all relevant aspects of the solution process. Every detail of the interview should be recorded by the developer, either by transcripts or by audio; conflicts and resulting consequences have to be illuminated and resolved, additional aspects should be thoroughly discussed, until a final complete picture of the application knowledge base evolves. Naturally, as the development of the application proceeds and additional or new questions arise, it may become necessary for the application developer to hold additional interview sessions with the expert(s) to reiterate or amend the knowledge base.

The Limit States Analysis Module knowledge base has been solicited from two experts in the field of nonlinear finite element analysis. One of the experts that was interviewed is Wendelin H. Mueller, a full professor at Portland State University (PSU), who developed the Limit Nonlinear Analysis Program for Bonneville Power Administration (BPA). Wendelin H. Mueller is considered an expert in the field of nonlinear analysis using post-buckling member performance behavior. The second expert interviewed was Leon Kempner Jr., a professional engineer working at BPA, who is also considered to be an expert in the area of nonlinear finite element analysis and a supporter of the Limit States Analysis program.

Both of the above named experts have been extensively interviewed about the Limit States Nonlinear Analysis program, its complexity, advantages and disadvantages, possible traps for the user, and many other related aspects. Hypothetical and real cases were utilized throughout the interviews and discussions in an attempt to capture all the relevant facts, heuristics, and rules that were necessary to develop the Limit States Analysis Module application. It was necessary to reiterate and amend the knowledge base and the expert system application many times until a satisfactory solution had been found. Full support had been given by both experts during the development phase, which was essential for the successful completion of the project, and was immensely appreciated by the developer of the application.

In addition, information and knowledge had been gathered and added to the knowledge base through other methods. Literature was consulted on the subject of nonlinear analysis using post-buckling member performance behavior and limit state analysis to obtain a general theoretical overview on the subjects. The source codes of the LIMIT B and CURVEPLOT programs were inspected, solution algorithms were examined, possible error sources were determined and noted. Last, a parametric study was performed using the LIMIT B program to determine the influence of variations in the control parameters on the results obtained. Any new findings obtained through these methods were added to the knowledge base.

The knowledge engineering and expert system implementation has been performed by the author. The author is currently enrolled in the Systems Science Phd program at PSU. During the time period of the research the author eventually became an expert himself in the field of nonlinear analysis using post-buckling member performances and familiar with the expert system methodology.

DESCRIPTION OF THE KNOWLEDGE

Within the following sections, a description is given of the knowledge and expertise that has been formulated based on the cooperation of the previously mentioned experts in the field of nonlinear analysis using post-buckling member performance. The first portions of the section deal with the more deterministic parts of the expert knowledge, while the later portions will describe the more intuitive aspects of the nonlinear analysis using the new LIMIT ES program.

The Limit States Analysis Module checks a variety of parameters and conditions of the LIMIT ES data input file. The program checks that the specified data input file for the LIMIT ES program exists, and that the file resides in the proper subdirectory. In addition, the program checks that all necessary data is specified in the file in its correct row and column format. Any deviations from the proper format are noted and documented as warnings in the pre-processor output file.

The pre-processor checks that none of the title lines contained in the input file exceed 70 characters, and that there are not more than 12 title lines overall. This is a necessary requirement since it will guarantee compatibility with the various editor modules. It also ensures that none of the key recognition characters used to designate the individual data blocks in a LIMIT ES input file are found on the first column of any of the title lines contained in the data block. If any of the key recognition characters is used within any of the first columns of the title lines the LIMIT ES program will attempt to read in the corresponding input data. This will result in the termination of the analysis due to a data format incompatibility. Any differences detected by the pre-processor are written as warnings to the pre-processor report file and the analysis terminates.

The pre-processor ensures that all of the control parameters that are specified in the data input file lie within a reasonable numerical range (i.e. that the modulus value or load multiplier increment specified are not negative, that the convergence criteria specified are not numerically smaller than what the accuracy of the data input permits, etc). If the modulus of elasticity is zero, the LIMIT ES program will terminate abnormally since a division by zero is not permissible in the solution algorithm. A negative value for the modulus of elasticity will cause the stiffness matrix of the structural system to be singular. A singular stiffness matrix can not be solved for a unique solution of values for all unknown variables, therefore rendering the results obtained by the analysis inaccurate. If the modulus value does not lie within the range of modulus values that are characteristic for structural steel members, the pre-processor will advise the user about the fact that the experimentally obtained database contained in the CURVEPLOT program will be violated. Any abnormalities will be documented in the pre-processor report file.

The pre-processor checks the joint data block of the input file that contains the information about the specified joints and their respective coordinates. It will flag any joints that have been specified twice (i.e. have an identical joint label) or have the label zero. Furthermore, the pre-processor will detect any joints that have been specified with identical numerical values for their coordinates in all three dimensions (i.e. two points that are located in the same position in space). This check is performed by a sophisticated vector algebra routine that is contained in the pre-processor module. It also ensures that every one of the joints that is defined in the joint data block is specified at least once in the member information data block as either a starting or ending joint of a member (i.e. there are no unconnected joints specified in the data input file which would cause LIMIT ES to calculate an inaccurate collapse load factor due to the instability). Any discrepancies are noted and added to the pre-processor report file.

The pre-processor performs a variety of checks for the member information data block. It checks that none of the members that are defined have identical starting and ending joints (i.e. two or more members are connected parallel to each other to the same two joints). Furthermore, the pre-processor will flag any members which have a starting and ending joint specified that are not distinct (i.e. both the starting and ending joint have the same joint label which would result in a joint to joint length of the member equal to zero). In addition, the pre-processor ensures that all of the joints that are used to specify the connectivity of the members exist and have been defined within the joint data block of the data input file. Any discrepancies discovered by the pre-processor are noted as warnings in the pre-processor file.

The program also checks that none of the members has a cross-sectional area that is less than or equal to zero; that the value specified for the KL/R ratio lies within the range from 60 to 300; and that none of the tension or compression capacities specified are less than or equal to zero. In addition, it will check that none of the compression capacities specified exceed their respective tension capacities. If the cross-sectional area of a member is specified to be equal to or less than zero the LIMIT ES will terminate because of a division by zero in the solution algorithm. Since the post-buckling member performance database was derived from tests on angles with a KL/R ratio of 60 to 300, the program will issue a warning to the user if the KL/R ratio boundaries of the database are exceeded. If any of the tension or compression capacities are specified to be equal to or less than zero the LIMIT ES program will not be able to calculate a first good run. In addition, the tension capacity of a member usually always exceeds the compression capacity if end connection effects are neglected. Any deviations are documented as warnings or comments in the evaluation report file.

The pre-processor also determines that no loads are applied at joints that possess an out of plane instability. A complex vector cross-product algorithm in combination with a dot-product algorithm searches for all joints that have members connected that lie within one plane. If all of the members attached to a joint lie within one plane, the algorithm will flag the joint as unstable, notify the user about the existing joint instability, and not allow the user to apply a load to this particular joint. In addition, a search mechanism checks that no loads are applied at joints that are not defined in the joint data block of the input file. The search mechanism notifies the user about any violations. Joints that possess an out of plane instability are joints that are connected to members that all lie within a plane (i.e. joints that are free to displace in the direction orthogonal to the plane made up of the attached members). These members are unstable in the out of plane direction since they do not have unit vector components in all three dimensions. If loads are applied at these unstable joints LIMIT ES will do one of two things:

- If artificial restraints are assigned, the loads applied to the unstable joints will be countered by soft springs which will stabilize the structural system. The springs will absorb a portion of the applied loads which will cause the collapse load factor calculated by the LIMIT ES program to be too high and may also produce excessive deflections. At the same time, LIMIT ES will calculate support reactions that are too small since some components of the loads are drawn into the soft spring supports (i.e. the sum of all applied loads will not equal the sum of all support reactions).

- If artificial restraints are not assigned, the loads applied to the unstable joints can not be resisted by the structural system allowing the unstable joint to displace excessively. Consequently, LIMIT ES will calculate joint displacements that are inaccurate. The resulting calculated collapse load factor will not be valid.

In addition, the pre-processor will ensure that each joint is connected to at least two members with distinct stiffness components (i.e. it determines all the joints that are intermediate). Intermediate joints are unstable joints that are connected to exactly two members that have identical unit vectors.

All of the dead loads applied to a structure usually have the same load direction (i.e. the direction the pull of gravity is assumed to exert influence). The pre-processor checks that all of the specified dead load components are applied in the same direction, and that none of the dead load components is applied to a joint that has not been specified in the joint data block. If a load is applied at a joint that is not specified in the joint data block, the LIMIT ES program will terminate runtime execution. In addition, the pre-processor ensures that none of the dead load components is applied to a joint that is unstable. Any discrepancies are documented as a warning in the pre-processor report file.

The pre-processor checks that none of the specified joint loads are duplicate or that joint loads are specified at joints that are not defined in the joint data block. It will flag any joint that has joint loads applied more than once in the same direction. If a load is applied at a joint that is not specified in the joint data block, the LIMIT ES program will terminate runtime execution. In addition, the pre-processor ensures that none of the joint load components is applied to a joint that is unstable. Any inconsistencies with the standard format are documented in the pre-processor report file.

The pre-processor determines if the structure that is to be analyzed is a 2-dimensional or 3-dimensional system. For both cases the pre-processor checks that enough independent specified deflections that support the boundary field have been defined for each of the orthogonal directions to guarantee the stability of the structure. If the number of specified deflections is incorrect the LIMIT ES program will do one of the following:

- If artificial restraints have not been assigned and the number of specified deflections is too small, the LIMIT ES program will not be able to complete a first good run and terminate the analysis process.

- If artificial restraints have been assigned and the number of specified deflections is too small, the LIMIT ES program will automatically assign artificial soft spring restraints to the unstable joints. If a starting load multiplier has been specified close to zero, the LIMIT ES program will calculate a collapse load factor that is not valid. If a starting load multiplier has been specified larger than zero, the LIMIT ES program may or may not be able to calculate a valid collapse load factor.

In addition, the pre-processor ensures that specified deflections have been defined for distinct orthogonal directions at each joint, and that no duplicate specified deflections exist. A vector algebra algorithm that uses a combination of cross- and dot products checks the validity of the specified directions and flags any violations. Any discrepancies found are noted and written into the pre-processor report file.

Some of the knowledge solicited from the experts indicated that changes had to be made to the original LIMIT B program to make it more adept to the development of the Limit States Analysis Module. It was recognized, that major modifications had to be made to the original LIMIT B program to adapt it to its new role within the Limit States Analysis Module. The changes that were made to the LIMIT B program are documented in the new LIMIT user manual and are described within the following paragraphs.

Limit ES was changed to allow the user to perform a checkrun with the specified data input file. In a checkrun LIMIT ES will read in all relevant data from a data input file and create a checkrun output file. The checkrun output file provides an input echo of the data input file together with an input echo of the post-buckling model performance curves. However, LIMIT ES will not calculate any member forces or joint displacements in the checkrun mode. The ability of LIMIT ES to perform a checkrun provides the user with another tool to verify the accuracy of the structural model. Once a checkrun has been performed the user is able to view the pre-processor report, use the expert input evaluation option, use the graphical modules, and use any of the screen editors of the Limit States Analysis Module program. This added feature allows the user to refine and optimize the nonlinear analysis using the available options of the Limit States Analysis Module previous to a full numerical analysis.

LIMIT ES was also changed to produce more output than the original LIMIT B version. The post-buckling model performance curves are included in the analysis output file together with an improved, more descriptive echo of the control parameters. In addition, checksums are created which add up columns of input data (i.e. sum of all areas, joint coordinates, joint labels, KL/R ratios, etc.) to provide the user with an additional tool to verify the accuracy of the input data file. Although these checksums are meaningless, they provide a reliable and rapid method to determine if any changes have been made to the data input file. If changes are discovered, the control program will notify the user of the Limit States Analysis Module. Furthermore, the new LIMIT ES program produces a runtime output file that is created during the execution of the analysis. The runtime log keeps track of the progress of the analysis and will also document any runtime execution statements. This feature allows the user to determine the cause of any runtime errors and to make corrections to alleviate the problem.

The original LIMIT B program terminated the analysis process whenever one member that did not have a model performance curve specified became overstressed. Many subsequent analyses had to be performed until all of the relevant members were determined and had been assigned post-buckling member performance curves. It was decided to eliminate this inefficiency by changing the nonlinear analysis methodology associated with the program. The significant change that was made to the program solution algorithm allows the program to determine all relevant and potentially critical members within one analysis. The LIMIT ES program is able to automatically assign a default post-buckling member performance curve once a member becomes overstressed (i.e. the analysis process does not stop once a member becomes overstressed). The LIMIT ES program is therefore now able to determine for the user all of the members that govern the nonlinear analysis which need to have member performance curves assigned to them within one analysis.

The LIMIT ES program uses a different and expanded set of post-buckling member performance curves than the original LIMIT B. The program attempts to qualify the selection of the appropriate post-buckling member performance curve contained in the LIMIT ES data input file for the user. A variety of factors influence the selection of the appropriate member performance curve.

One of the parameters that exerts the most influence on the post-buckling behavior of a member is the member's KL/R ratio. The Limit States Analysis Module member performance database was developed from physical member tests of single and double steel angles. Tests were performed at PSU for members with KL/R ratios in the range of 60 to 300 for single angles (5, 26), and 60 to 210 for double angles (28). Based upon new findings from the physical member tests new model performance curves were developed and incorporated into the program. The set of the model performance curves was expanded and modified to reflect the influences of load eccentricities, member end conditions, and the existence of intermediate lap and butt joints on the post-buckling performance of members.

As mentioned previously, the shape of the post-buckling member performance curve is directly affected by the KL/R ratio. The program assists the user in the selection of the appropriate member performance in a number of ways. It displays both, the selected model performance curve and the true member performance curve simultaneously on screen for the user. The normalized model post-buckling performance curve is adjusted by the program in a number of ways. First, the program adjusts the model curve to reflect the fact that the capacity of the model curve equals the capacity of the predicted performance curve. Second, the program scales the model curve based on the difference between the normalized displacements and the true displacements. This ensures that both curves, the model curve and the predicted performance curve use true displacements and loads. In addition, the program aids the user in the process of adjusting the model curve based on the geometric properties and existing end conditions of the member. This last adjustment is done to eliminate the influence of bowing on the after failure performance of the member caused by the difference between the effective and the joint to joint length. Once all the adjustments have been made by the program, the user is able to either accept the model performance curve displayed by the program or to reiterate through the process to choose a different model curve.

In order to account for different end conditions, the program changes the shape of the displayed predicted performance curve to reflect the differences. Again, the user is able to either accept the model performance curve shown by the program or to choose a different one.

All of the above mentioned tasks associated with the selection of the appropriate model performance curve previously to the development of the program had to be done by the user for every member in the tower that showed to be overstressed in an elastic analysis. The normal sequence of events that had to be done previous to the program included the calculation of the ultimate capacity of the member as a function of yield stress, end conditions, brace and stitch points, KL/R length and joint to joint length, etc.; the shifting of the two normalized curves in such fashion that the normalized ultimate capacities of both curves are identical; the shift of the true normalized curve along the horizontal axis to account for the effects of the difference between the KL/R and joint to joint length; and finally the denormalization of the normalized model performance curve to true load and deflection values which are used in the LIMIT ES solution process. The program successfully eliminates numerical errors that can result from the various adjustment procedures since it always considers all possible influences on a member's post-buckling behavior.

Once the nonlinear analysis has been completed, the program aids the user in the final verification of the appropriateness of the selected model performance curves. The program displays the selected model performance curve together with the predicted performance curve and also displays a vertical line at the point where the member is performing. If the correlation of all the model and predicted performance curves is close everywhere and at the point of the members' individual performances, the selection of the set of model curves was appropriate and the analysis is complete.

The program also aids the user in the selection of the appropriate control parameters. Slight variations in some of the control parameters will cause LIMIT ES to calculate different collapse load factors. There is no unique set of control parameters that will produce accurate results. The most appropriate set of control parameters depends much on the problem context and the structural system to be analyzed.

The program checks the user selected control parameters and will document any comments or warnings resulting from the evaluation in the input evaluation report. In addition, it will make recommendations on how to improve the selection of the control parameters for the specified data input file. The expert system arrives at these recommendations in a number of ways such as:

- If the analysis to be performed is an elastic analysis, the evaluation module will simply check all data input for validity and consistency and make sure that all control parameters are set to the most appropriate default values (i.e. the LIMIT ES program performs an elastic analysis only to validate the structural model; it's main purpose however is to perform a nonlinear analysis using post-buckling member performance curves).

- If the analysis to be performed is a bilinear analysis the evaluation module will check that all necessary control parameters have been selected and specified. The program will then proceed to evaluate the data input file and suggest appropriate values (i.e. if possible) for the limiting KL/R ratio, if artificial restraints should be assigned or not, the default bilinear curve number that should be used in the analysis, make a recommendation for the starting load multiplier and the load multiplier increment, and suggest recommended convergence criteria.

- If the analysis to be performed is an analysis using normalized or actual member performance curves the evaluation module will analyze the data input file as described above. Pending the results of the input evaluation, the program will suggest appropriate control parameters that are more suited to the normalized or actual analysis.

Independent of the type of analysis that is to be performed, the input evaluation will estimate the chance of success of calculating a valid collapse load factor with the specified input data file. The chance of success is calculated as a percent chance based on 100 percent and is documented in the input evaluation report. The percent chance of success reported by the input evaluation allows the user to obtain an idea about the confidence he\she should place in the results of the LIMIT ES nonlinear analysis.

Last, the program aids the user in the interpretation of the LIMIT ES analysis output file. The output evaluation will verify the appropriateness of the analysis by interpreting certain parameters contained in the output file. Probabilities are assigned to the evaluated results and the collapse load factor calculated which are reported in the output evaluation report.

Depending upon the type of analysis that was performed the program helps the user identify the critical members of the structural system (i.e. the members that are involved in causing the collapse of the structural system). The output evaluation attempts to define the most critical member (i.e. the one particular member that is responsible for the collapse of the structure). In addition, the output evaluation will calculate the probability that the particular member identified as the most critical member is the member that causes structural failure.

Finally, as described previously in the input evaluation, the program advises the user on how to improve the analysis further by making recommendations on how to improve the control parameters based on the results of the analysis.

STRUCTURING AND ENCODING THE KNOWLEDGE

The knowledge base was divided into two major components. One component included all the knowledge and experiences that were of a somewhat intuitive nature and depended to a large extent upon the particular problem context. It was decided, that this portion of the knowledge base was going to be encoded with the help of the development shell NEXPERT OBJECT. The other portion of the knowledge base proved to be of a more deterministic nature and was less suited to be encoded with the development tool. The decision was made to use a traditional third-level programming language (MICROSOFT PROBASIC 7.1) to write a numerical pre- and post-processor that would contain and apply this portion of the knowledge.

The deterministic portion of the knowledge base, described previously, was programmed as a series of IF-THEN statements in combination with numerical algorithms. The numerical algorithms contained in the pre-processor are typically used to check a portion of the input data to determine if a certain condition exists (i.e. such as checking if an out of plane instability of a joint with an applied joint load exists). Pending the results, specific flags are set which determine what comments will be written to the pre-processor report. Comments contained in the pre-processor report indicate to the user the precise location and nature of any problems encountered.

The knowledge contained in the pre-processor program is organized in the classical hierarchical programming style as a series of prioritized IF-THEN statements. The purpose or goal of the pre-processor is to ensure that the user specified input file will produce a successful analysis. A successful analysis is defined as an analysis that will produce results using the LIMIT ES program and not terminate unexpectedly. However, the pre-processor does not guarantee that the results calculated are correct. The pre-processor therefore serves essentially as an analysis verification tool prior to the actual analysis.

Next, it was decided to divide the intuitive portion of the knowledge base into two separate components. One component contains all of the knowledge that pertains to the preparation of a LIMIT ES input data file, and the selection of the appropriate member performance curves and control parameters. This portion of the knowledge base was used to develop the input evaluation module. It was felt, that it would be advantageous for the user to obtain some feedback from the input evaluation before actually running the LIMIT ES analysis so as not to waste time in performing an analysis with an inappropriate data input file. The input evaluation therefore provides a type of input data validation prior to running the actual analysis. Due to the nature of a nonlinear analysis using the LIMIT ES program the validation will be incomplete until the analysis has been run, but it will allow the user to arrive at the final solution much faster.

The second component contains all of the knowledge that is more suited for the interpretation of the analysis output and the determination of the next logical step on how to proceed with the refinement of the LIMIT ES analysis. This portion of the knowledge base was used to develop the output evaluation module. The output evaluation evaluates both, the input data and the results calculated by the LIMIT ES program. Based on the results of the output evaluation, the application will provide the user with an indication about the level of confidence he\she should have in the results calculated. In addition, the output evaluation will make specific recommendations on how the user should proceed with the analysis.

Some of the knowledge that was solicited from the human experts could not be categorized to either pertaining to the input or output evaluation modules. Therefore, it was decided, to include this expertise in both, the input evaluation module and the output evaluation module.

The input evaluation was programmed using the NEXPERT OBJECT development shell. The ultimate hypothesis (goal) of the knowledge base is to evaluate the information contained in the user specified input file. Once the ultimate hypothesis is evaluated to be true, the input evaluation is complete, the inference process stops, and the control of the analysis is returned to the user. Therefore, the input evaluation application acts in the role of a completely independent expert advisor to the user. The input evaluation knowledge base is subdivided into five separate knowledge islands that are connected with each other through context links. One of the knowledge islands contains all the rules that pertain to reading in all of the information contained in the LIMIT ES input file and writing all of the results (i.e. comments and recommendations) to the input evaluation report. It is this knowledge island whose hypothesis is placed on the agenda of the inference engine once the input evaluation process is started. Since there is only one ultimate hypothesis (i.e. input evaluation completed), it was possible to have the application finish the evaluation process completely independent from any interactive user response. This ensures that the knowledge base can not accidentally be changed by the user. Each one of the other four knowledge islands contains the information and rules that apply to each of the types of analyses (i.e. elastic, bilinear, normalized, actual) that LIMIT ES can perform. The knowledge pertaining to the different types of analyses was divided into four independent knowledge islands to save computational time by eliminating the portions of the expertise that are not necessary to arrive at the appropriate set of conclusions and recommendations.

The input evaluation utilizes backward chaining throughout the object and rule structure until it determines which rules have to be fired. The rules that are contained in each of the knowledge islands are organized into logical frames which contain related information. By process of elimination, based on a prioritization scheme, the inference engine will only fire one rule within each of the frames until it determines the complete set of conclusions and recommendations that will be written to the input evaluation report. In addition, certainty levels are assigned to the firing of each individual rule within each frame. These certainty levels are used to give the user an indication about the level of confidence he\she should place in the results obtained by the LIMIT ES analysis with the user specified input data file. The rules were organized in this fashion to ensure that no conflicting recommendations would be made by the input evaluation knowledge base.

There are two types of structures contained in the input evaluation. One of the structures is the rule structure which guides the inference engine on how to process the information stored in the working memory. The other structure is the object structure that indicates to the inference engine on how the classes, objects, properties, and slots are related to each other. The input evaluation application has a total of eight classes. The properties attached to each of these classes define what properties the objects can have if the inheritance proceeds downward from classes to objects. Typically, inheritance strategies are specified in such manner that an object can not have any properties other than the ones defined for the parent class. Almost all of the objects in the input evaluation are dynamically created during the inference process and therefore inherit their properties from their parent class(es). The default inheritance method that was selected in the input evaluation is "Inheritance Downward" (i.e. classes inherit properties from their parent class(es), objects inherit their properties from their parent object(s) or class(es)). If an object has two or more parent classes assigned to it, the inference engine will determine through its conflict resolution process which properties will be inherited downward. The default conflict resolution scheme used in the input evaluation is dependent upon which hypothesis is put on the agenda first (i.e. inheritance of properties occurs whenever necessary in order to evaluate the hypothesis of a certain rule). The objects are created during runtime execution, their properties are inherited from their parent class(es), and all of the input information is stored in the slot values of the properties attached to the dynamic objects. The objects are created dynamically because the lengths of the records that are read in by the input evaluation vary for each individual input file. Using this method ensures that the input evaluation does not create any objects that contain any unknown values (i.e. comparable to defining a dimension for a traditional array of variables which may or may not be filled with values) which reduces the amount of memory that needs to be allocated. In an expert system it is possible for a variable to take the value "unknown" which can cause the expert system to report an inaccurate conclusion or make a bad recommendation unless the value has been anticipated by the programmer.

The output evaluation has been structured and programmed in much the same way as the input evaluation Again, the knowledge is structured into five individual knowledge islands that are connected to each other by context links. The main knowledge island controls the inference process and contains all the rules that pertain to retrieving all of the information contained in the LIMIT ES input and output file. The ultimate hypothesis (goal) of the knowledge base is to evaluate the results obtained by the LIMIT ES analysis. Once the ultimate hypothesis is evaluated to be true, the output evaluation is complete, the inference process stops, and the control of the program is returned to the user. Again, as it was done in the input evaluation, the knowledge pertaining to the different types of analyses was divided into four independent knowledge islands. This was done to save computational time because it eliminates the portions of the expertise that do not need to be considered to arrive at the appropriate set of conclusions and recommendations.

Both of the knowledge bases were tested and verified for all possible instances. Any conflicts and programming bugs were resolved and corrected. The final versions of the input and output evaluation modules were then compiled within the NEXPERT development shell to form the final standalone runtime modules that are used within the Limit States Analysis Module.

As the final step, it was necessary to develop the runtime definition files that would allow the evaluations to be run just like an executable file rather than from within NEXPERT's development shell. NEXPERT applications can be used either within the MS Windows environment or as a character based MS DOS application. If the application is used within the MS Windows environment the user is able to change any of the components contained in the rule and object structure of the application. The disadvantage of running the application in the MS Windows environment is that the inexperienced user could change the knowledge base accidentally rendering the application useless. If the application is run as a standalone runtime executable in the MS DOS environment, the knowledge base is compiled and can not be changed accidentally by the inexperienced user. It was decided to run the evaluation applications as standalone runtime executables in the MS DOS environment to first, minimize the risk of accidental editing, and second, to be able to run the LIMIT ES program in an environment that allows it to utilize extended and expanded memory.

Two runtime definition files were created, one for each of the evaluation applications. A runtime definition file essentially contains the information that will start the NEXPERT OBJECT inference process, retrieve the appropriate knowledge base, name the hypothesis to be placed on the agenda for evaluation, and contains the name of the output format file that is to be used to create the evaluation report. Once the value of the hypothesis is evaluated, the runtime definition file will terminate the inference engine and return control to the user.

INTERACTION OF THE EVALUATIONS AND THE LIMIT NONLINEAR ANALYSIS PROGRAM

All the relevant data for the input or output evaluation is read from the LIMIT ES input and output files. Therefore, user interaction is not required during the runtime execution of the input or output evaluation. The evaluations are completely self-contained, and all conclusions, recommendations, comments, and warnings are documented in a report file. The report file is displayed on the computer screen to the user at the end of the inference session. The same report file is automatically saved and can be recalled by the user at any time later in the analysis.

All relevant data for the evaluations is read into NEXPERT's working memory through NEXPERT's database interface from the LIMIT ES input and output files. The objects that are created dynamically during the runtime execution of the evaluations will inherit their properties from the parent class(es). All the information that is read from the files is stored in the slots of the properties associated with these dynamic objects. Once the session is started, the default hypothesis is put on the agenda, and the inference process begins.

The inference engine backward chains until it determines the set of rules that needs to be evaluated as true to support the hypothesis on the agenda. It then proceeds to evaluate all relevant rules completely in its forward chaining process until it reaches and evaluates the final hypothesis. At this point the inference engine shuts down and the runtime definition file takes control and writes the evaluation report. The evaluation report is then displayed on screen until the user hits the optional <END> key. Once the evaluation report is exited, the control returns to the Limit States Analysis Module main menu and the user has the choice to print or view the evaluation report through the file viewer or to proceed in any other way.

The input or output evaluation reports will be available to the user until the LIMIT ES data input file associated with the evaluation reports is changed. If the data input file is changed from anywhere within the Limit States Analysis Module menus or screen editors, the program will delete all associated output files and evaluation reports. Therefore, it is not possible for an evaluation or output file to exist for a specific data input file unless they pertain to the exact same model and are up-to-date.

Since the evaluations are completely separated from the other components of the Limit States Analysis Module, it will be easy to update the knowledge bases whenever new information is obtained. As a result of this modular design it is possible to change and update any of the other components of the Limit States Analysis Module without invalidating the input and output evaluation modules. In addition, since the evaluations are separated from the Limit States Analysis Module the experienced user may or may not choose to use the evaluations (i.e. the use of the evaluations is optional and can be omitted by the user experienced with the LIMIT ES program).

VERIFICATION AND VALIDATION OF THE LIMIT STATES ANALYSIS MODULE

The Limit States Analysis Module was verified using a variety of approaches. Verification is defined in this document as the process of checking the program in such manner that it is ensured to do everything as expected. First, the program was debugged and Beta tested by three different people independently. Two of the people that were testing the program had not been involved in the development of the program modules and were asked to also comment on the appropriateness of the menus, editors, help files, etc, and asked to document any problems or questions. At the end of the testing period notes were compared, evaluated, and necessary changes were discussed. Changes were implemented and the Limit States Analysis Module was retested. This cycle of testing and re-evaluating continued until all aspects of the program proved satisfactory.

Next, all the read and write statements used in the Limit State Expert System and in the LIMIT ES programming code were tested. This included testing of all the screen editors in which data of the input file can be entered or changed. Hypothetical cases were developed to verify the capabilities of the graphic routines and last changes were made wherever necessary.

Once the Limit State Expert System had been verified completely and was deemed satisfactory, a validation process was designed to test the Limit States Analysis Module. Validation is defined in this document as the process of determining if the results calculated are appropriate and reasonable. Two analyses of different towers were run for which the collapse load factor had been previously determined in full scale tower load tests. The results of these two analyses were reported by Ostendorp (25) and Hesse (24). The analyses were performed in order to establish performance values, such as the time required for a nonlinear analysis, appropriateness of the expert evaluations, and to check the accuracy of the collapse load factor calculated by the program. The calculated collapse load factors were compared to the known values that were obtained from full scale tower tests and were found to compare well (i.e the results did not differ by more than ten percent).

Finally, the validity of the Limit States Analysis Module was checked further with the help of two program testers that had no previous exposure to the nonlinear analysis principles using post-buckling member performance behavior. The two testers were asked to perform a nonlinear analysis for a different tower than the one mentioned previously. Both program testers received the same identical elastic Limit State Expert System data input file. The testers were asked to perform the analysis independently from each other and to take notes as they went through the nonlinear analysis process. They were encouraged to use the expert input and output evaluation whenever necessary and were also advised about the existence of the on-line LIMIT ES user (22) and modeling (23) manual.

Both testers were able to complete the analysis within six to eight hours and reported calculated collapse load factors that were within two percent of each other. In addition, both of the testers determined the same members to be the critical members. The testers differed somewhat in their choice of the post-buckling member performance curves for some of the members, which explained the difference in the calculated collapse load factors. It is possible for two people to choose different nonlinear post-buckling model performance curves and can be expected frequently. The choice of the most appropriate curve is somewhat arbitrary, and there can be one or more appropriate model performance curves for a specific member. The choice of the most appropriate model performance curve depends much upon the geometric characteristics of the member and the perception of the user.

In addition, two engineers associated with BC HYDRO, a canadian power utility company, performed a nonlinear analysis for the same tower assisted by one of the experts. The expert that assisted the two engineers is Leon Kempner. The two engineers had no prior exposure to performing a nonlinear analysis using the LIMIT ES program with post-buckling member performances. Both of the engineers received the same elastic Limit States Analysis Module input data file as the two student testers. The two engineers determined collapse load factors that were close to the collapse load factor determined by the student testers (i.e. all of the collapse load factors were within five percent of each other). In addition, the two engineers identified the same member to be the critical member that causes collapse of the structure as the other testers.

In conclusion, it can be stated that the Limit States Analysis Module allows an inexperienced user to perform a valid and reasonably accurate nonlinear analysis using post-buckling member behavior. The combination of expert advice coupled with a powerful numerical analysis program provides a reliable, sufficiently accurate, and user-friendly software package.

IMPLEMENTATION OF THE LIMIT STATES ANALYSIS MODULE IN DESIGN AND ANALYSIS

Traditionally, new towers were designed in the usual manner with an elastic analysis tool such as the TOWER - Tower Analysis and Design program for all the possible load cases that could occur. The design would then be refined further using the elastic analysis tool until a satisfactory solution was determined. The final design of a tower is always based upon a factor of safety with respect to the idealized elastic capacity. The factor of safety that is used by BPA equals unity (i.e. the ratio of load demand to idealized structural capacity equals 1.0). It was suspected, that the true capacity of a tower design was somewhat higher than the elastic analysis was able to predict. Full scale tower tests indicated that the true capacities of the towers were larger than the elastic analyses would predict.

The original LIMIT B program was developed in an attempt to obtain more information about the true capacity of a tower. Full scale tower tests were performed to verify the accuracy of the LIMIT B program and the results obtained compared very well with the test results. However, the LIMIT B program's purpose originally was to be used as a research tool only.

With the development of the Limit States Analysis Module it has now become possible for the average engineer (i.e. an engineer inexperienced with the concepts of a nonlinear analysis using post-buckling member performance curves) to perform a nonlinear analysis with the LIMIT ES program using post-buckling member performance behavior in a production design environment. A typical Limit States Analysis Module analysis will now at the most take a few hours for each load case and can be run by an engineer who is not an expert in the field of nonlinear analysis. Consequently, designs can be analyzed to establish their true reserve capacity. Knowing the reserve capacity of a tower will increase the engineer's confidence in his\her assumptions and promote designs that are more economical.

Originally it was intended to install the Limit States Analysis Module on the VAX system at BPA. Due to technological changes during the research and development phase of the Limit States Analysis Module it was decided to change from the VAX environment to the IBM PC environment.

It was decided to purchase two 80486 IBM personal computer systems that would be used as the designated nonlinear analysis platforms for the Limit States Analysis Module. Once the program system had been installed, it was determined that the LIMIT ES program experienced difficulties on these computer systems during the execution phase of the LIMIT ES FORTRAN 77 code. After consultation with the software distributor, it was determined that the compiler used in the development of the LIMIT ES program was not able to utilize the advanced architecture of the new 80486 microprocessor. A different FORTRAN 77 compiler, developed specifically for the 80486 micro-processor, was then purchased by BPA, and the LIMIT ES program was recompiled. No other changes were made to the programming code at this point.

CHAPTER V

SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

SUMMARY

A program has been designed and developed that combines all the necessary components to allow an inexperienced user to perform a nonlinear limit states analysis using post-buckling member performance behavior. This program is the Limit States Analysis Module. The following research and development has been performed in order to develop the Limit States Analysis Module:

- The experience and knowledge of two experts in the field of nonlinear analysis using post-buckling member performance behavior has been solicited to create the Limit States Analysis Module knowledge base.

- The fuzzy portion of the knowledge base was examined and structured to allow the developer to encode the expertise with the help of a development shell. The knowledge base was divided into two portions, the knowledge associated with the preparation of the analysis input file, and the knowledge associated with the interpretation of the analysis output files.

- The input and output evaluation knowledge bases were encoded with the help of the NEXPERT OBJECT development software package.

- The input and output evaluation knowledge bases were verified and validated through extensive testing. Last minute changes were made and the final format of the input and output evaluation reports was developed.

- Changes were made to the original LIMIT B analysis program. Changes included converting the LIMIT B program so that it would be able to operate on an IBM PC within the MS DOS environment and take advantage of available extended memory and co-processors such as a WEITEK. In addition, changes were made to the input and output file structure of LIMIT B that allowed the Limit States Analysis Module to obtain more detailed information. Control parameters were added to the program to facilitate the use of the newly developed solution algorithm.

- A member test program was developed. Steel angle sizes and shapes were selected based on a histogram. The histogram that was used is based on the frequency of use of certain steel angle types in existing tower designs.

- A test setup was developed to test the selected steel angle members in compression. A computer data acquisition system was developed to pre- and post-process the test results.

- A regression analysis was performed to develop a new post-buckling member performance database for single angles with equal legs, single angles with unequal legs, and double angles with equal or unequal legs.

- The CURVEPLOT post - buckling member performance behavior database was changed to incorporate the new findings obtained from the physical member tests performed at Portland State University (26, 27, 28).

- New post-buckling member performance model curves were developed based on the changes that were made to the CURVEPLOT database. The new model curves were then verified and validated for fit, form, and function by performing a number of LIMIT ES analyses using post-buckling member performances.

- A new nonlinear analysis algorithm was developed using the LIMIT ES program. The new algorithm allows the user to perform a complete nonlinear analysis using the LIMIT ES program with post-buckling member performance curves in a fraction of the time required previously.

- A load history graphics module was designed and incorporated into the Limit States Analysis Module that allows the user to visually identify critical member performances through the loading stage of the analysis process.

- A graphical tower plotting routine was designed and incorporated into the Limit States Analysis Module that allows the user to visually verify the structural model that is being analyzed. Extended options include the display of critical members, zooming in and out of the display, rotation about any of the three axes, display of joint or member labels, identification of dead and live loads, etc.

- The CURVEPLOT post - buckling member performance graphics module was redesigned and incorporated into the Limit States Analysis Module to include the new experimental findings. Added options include new member performance model curves, display of member information, the ability to select new curves on screen, a better representation of true and model performance curves, and the ability to view the LIMIT ES output file.

- A user interface was designed and developed that includes a main menu, input and output file directory, input and output menu, on-line LIMIT ES user and modeling manual, full text editor and file viewer, bandwidth optimization menu, complete input data editors, and on-line help screens.

- Created and incorporated on-line ASCII versions of the LIMIT user and modeling manuals. Manuals can be viewed on screen from within the main menu.

- A data input file pre-processor was designed and added to the Limit States Analysis Module that checks the input file for format, basic modeling errors, and data ranges. The pre-processor applies the portion of the knowledge base to the analysis that is of a more deterministic nature. It checks for missing joints, multiple joints with identical coordinates, members with areas less than or equal to zero, incorrectly applied loads, etc.

- On-line help files were developed and incorporated into the Limit States Analysis Module. Help is available on any subject of the Limit States Analysis Module that requires user interaction.

- Verification of the Limit States Analysis Module. An analysis was run with the program by the author and one other person. Results obtained through the analyses were compared to results that were obtained from full scale tower tests.

- The Limit States Expert System was installed on computer platforms at BPA. A presentation was given to the sponsors at BPA that illustrated the capabilities of the program.

- The capabilities of the Limit States Expert System were tested and validated within the production design environment at BPA. A tower design was analyzed independently by three different parties. One of the parties was assisted by an expert. Final results were compared to results obtained by full scale load testing of the tower. All three analyses produced calculated collapse load factors that did not differ by more than two percent and established identical collapse mechanisms (i.e. the collapse occurred always as a result of one specific member). The results calculated compared well with the results obtained from the full scale load test.

CONCLUSIONS

The successful development of the program resulted in a complete, working program application, that combines all of the components that are required to perform a nonlinear finite element analysis using post-buckling member performance into one program module. This program module is the Limit States Analysis Module.

The Limit States Analysis Module guides the inexperienced user through all the steps that are required to successfully perform a nonlinear finite element analysis, including the selection of the appropriate post-buckling member performance behavior. The Limit States Analysis Module is able to function in such capacity because of its elaborate built-in program control that will protect the user from most errors that usually can occur in a nonlinear analysis. The built-in program control has been achieved as a result of the use of numerical and graphical pre-processors, the incorporated advisory knowledge bases that evaluate the nonlinear analysis input and output, the use of extensive help screens, on-line program manuals that utilize graphical displays, tightly controlled screen editors for any of the required input data and control parameters, and an extensive net of data format and runtime error trapping. However, at the same time the Limit States Analysis Module is flexible enough to also allow the experienced user to perform an analysis efficiently without having to use all of the program options that the inexperienced user would use.

The graphical pre- and post-processors enable the user to verify the model visually and provide documentation that may be used within a written documentation such as an analysis report. A screen capture utility has been built into the Limit States Analysis Module that allows the user to capture any display that is seen on the screen into graphical bitmap files that can then be printed out and\or pasted into a report document after the analysis has been completed.

It has been determined, that the time necessary to perform a nonlinear finite element analysis with the original LIMIT B program code using post-buckling member performance curves can be drastically reduced if one uses the Limit States Analysis Module. Early comparisons show that utilizing the Limit States Analysis Module can reduce the time required for a LIMIT ES analysis by anywhere from 75 to 90 percent depending upon user experience with the necessary theoretical concepts and the program. The indicated time savings allow the LIMIT ES analysis program now to be used as a production tower design tool that complements BPA's standard TOWER - Tower Analysis and Design program.

With an increase in the number of nonlinear analyses performed by the inexperienced user with the Limit States Analysis Module, the user will become more and more familiar with, and confident in the nonlinear finite element analysis process that uses post-buckling member performance behavior. Therefore, the user will eventually become an expert in the field. Consequently, the program combines educational aspects with productive work, which eliminates much of the downtime usually associated with the formal training into a new technology. This, in return, will reduce the financial burden placed on a company, or in this case to be more specific the financial burden placed on BPA. At the same time, the Limit States Analysis Module reduces the work load of the resident human experts freeing them for other tasks. In addition, the Limit States Analysis Module preserves the resident engineers's expertise and knowledge for the future. Instead of losing the human expertise through retirement, the company will be able to benefit from the collected knowledge in the future, utilizing the investment made into training in an efficient manner.

The use of the Limit State Expert System improves the accuracy and consistency of the results obtained from a nonlinear analysis using post-buckling member performance behavior. This increases the confidence of the engineers in the analysis process and also produces more economical designs. Overall, the Limit States Analysis Module decreases the amount of time necessary to perform a nonlinear analysis that uses post-buckling member behavior, which increases the productivity of the individual, or section, or department, and results in economical savings to the company.

RECOMMENDATIONS

Future efforts should be directed towards the expansion of the Limit States Analysis Module in such manner that it would perform the actual design of the individual members. The actual design of the individual members should be based on the conventional stress or strength design methods currently used in the transmission tower design industry (20, 21).

The Limit States Analysis Module could also be expanded in such manner that it would be able to consider the nonlinear soil response on the foundations of the tower. A set of normalized model curves could be developed based on geotechnical data which would be used in a similar manner as the post-buckling member performance model curves. Nonlinear soil response model curves could then be assigned to each of the foundation joints for both, lateral and vertical movements. Tower structures could then be analyzed in even more detail since the site specific characteristics would be included in the determination of the collapse load factor.

The post-buckling member performance database of the CURVEPLOT program should be expanded. Additional tests should be performed on single steel angles, both equal leg sizes and unequal leg sizes, and double angles with a specified yield strength of 50 ksi. Many tower designs utilize steel angles with yield strengths of 36 and 50 ksi. No compression tests have been performed to assess the influence of the higher yield strength on the post-buckling behavior of the members. It is anticipated that the steel angles that have a higher yield strength will show a more brittle mode of failure, and this could significantly influence the magnitude of the collapse load factor calculated by the LIMIT ES program.

A test program should be developed that assesses the influence of joint rotational restraints on the post-buckling behavior of steel angles. Furthermore, tests should be performed on sub-structural assemblies to obtain more information about the interactions and the load flows that occur in transmission towers. Results of these tests could then be used to expand the CURVEPLOT member performance database and to enhance the knowledge base contained in the expert input and output evaluations.

In addition, it is feasible to expand the Limit State Expert System to include a linear and nonlinear probability based analysis method using post-buckling member performances and a reliability based design approach. The probability based analysis method could be used to establish failure as a function of individual member collapse as well as overall system collapse. Probabilities of failure could then be matched more evenly for each of the systems components and the overall system itself. The set of all possible collapse failure mechanisms could be determined through manipulation of the structural stiffness matrix. The collapse load mechanisms can be determined from all the instances when the structural stiffness matrix becomes singular by identifying the members or combination of members that would cause the singularity. On the other hand, incorporating a reliability based design would allow the designers to match the reliability of the system with the reliability of any of its components. This would enable the designer to establish a more uniform factor of safety for the system and any of its components.

REFERENCES

(1) Mueller, W. H., Pricket, S. L., Kempner, L. Jr (1988) "LIMIT: A Space Truss Analysis Program with Post-Buckling Member Behavior" Portland State University

(2) Bonneville Power Administration (1987) "Tower Analysis and Design" Bonneville Power Administration

(3) Miller, M. D., Mueller, W. H. (1986) "CURVE_PLOT.4 - Member Performance Database for LIMIT" Portland State University

(4) Mueller, W. H., Pricket, S. L., Kempner, L. Jr (1984) "Nonlinear Analysis and Full Scale Test of a Transmission Tower" Portland State University

(5) Mueller, W. H., Wagner, A. L. (1983) "Plastic Behavior of Steel Angle Columns" Portland State University

(6) Mackerle, J., Osborn K. (1988) "Expert Systems for Finite Element Analysis and Design Optimization - a Review" Journal of Eng. Comput., Volume 5

(7) Wriggers, P., Tarnow, N. (1989) "Interaktive Steuerung von Nichtlinearen Finite-Element-Algorithmen mittels eines Expertensystems" Der Bauingenieur, Volume 64

(8) Li, Cao (1989) "Pretap: An Expert System Pre-Processor for a Structural Analysis Program of Tall Buildings" Computers & Structures, Volume 33 No.3

(9) Forde, Bruce W. R., Stiemer, Siegfried F. (1989) "Knowledge-Based Control for Finite Element Analysis" Eng. with Comp., Volume 5

(10) Ramirez, Martin R., Belytschko, Ted (1989) "An Expert System for Setting Time Steps in Dynamic Finite Element Programs" Eng. with Comp., Volume 5

(11) American Society of Civil Engineers (1986) <u>Expert Systems in Civil Engineering</u> ASCE Conference Proceedings

(12) American Society of Civil Engineers (1987) <u>Expert Systems for Civil Engineers: Technology & Applications</u> ASCE Publication

(13) American Society of Civil Engineers (1989) Computing in Civil Engineering ASCE Conference Proceedings

(14) Prerau, David S. (1990) Developing and Managing Expert Systems Addison-Wesley Publication Company

(15) Smith, Erling A. (1984) "Space Truss Nonlinear Analysis" Journal of Structural Eng., Volume 110, No.4

(16) Tauchert, Theodore R. (1981) Energy Principles in Structural Mechanics Robert E. Krieger Publ. Company

(17) Chen, W. F., Lui, E. M. (1987) Structural Stability: Theory and Implementation Elsevier Publ. Co.

(18) Timoshenko, S. (1936) Theory of Elastic Stability McGraw-Hill Book Co.

(19) Salmon, Charles G., Johnson, John E. (1980) Steel Structures: Design and Behavior Harper & Row Publ. Co., 2 nd Edition

(20) American Institute of Steel Construction (1980) Manual of Steel Construction AISC, Eight Edition

(21) ASCE Manuals and Reports on Engineering Practice No. 52 (1982) Guide for Design of Steel Transmission Towers Published by ASCE, 2 nd Edition

(22) Mueller, W. H., Ostendorp, Markus, Kempner, L. Jr (1991) "LIMIT: A Nonlinear Three Dimensional Truss Analysis Program" Portland State University, 2 nd Edition

(23) Mueller, W. H., Kempner, L. Jr (1991) "Modeling Manual for a LIMIT Analysis" Portland State University

(24) Hesse, D. M., Mueller, W. H. (1991) "Case Study #1: Limit Analysis of the 52S Transmission Tower - Load Case S" Portland State University

(25) Ostendorp, Markus, Mueller, W. H. (1991) "Case Study #2: Limit Analysis of the 338M Transmission Tower - Load Case A" Portland State University

(26) Bathon, Leander A., Mueller W. H., Kempner, L. Jr (1990) "Post-Buckling Behavior of Single Steel Angles" Portland State University

(27) Robinson, J., Mueller, W. H. (1991) "Analytical Model of Post-Buckling Performance in Equal, Unequal, and Double Leg Steel Angles" Portland State University

(28) Huntley, Kenneth R., Mueller, W. H., Kempner, L. Jr (1991) "Post-Buckling Behavior of Double Angles, Lap and Butt Joined Angles, and Cross-Braced Members" Portland State University

(29) Van Den Broek, J. A. (1947) Theory of Limit Design John Wiley & Sons, Inc., New York

(30) Beedle, Lynn S. (1958) Plastic Design of Steel Frames John Wiley & Sons, Inc., New York

GLOSSARY OF TERMS

Agenda: The agenda is a mechanism by which events are scheduled to happen during a knowledge processing session. The agenda is a dynamic mechanism that provides the central transformation between the perception of events and the actions taken in response.

AI: (Artificial Intelligence) The science of making machines behave in a way that when done by human beings is regarded as intelligent behavior.

AI Paradigm: A mechanism that can be used to represent knowledge in an expert system program, such as rules, and frames.

AI Programming Language: A programming language specifically designed to be used in applications which attempt to imitate the human reasoning process.

Antecedent: The left hand side of an IF\THEN rule.

Backward chaining: A reasoning technique used in production rule systems which starts from a given goal, and works backward to find the hypotheses which need to exist in order to arrive at the goal.

Certainty factor: A quantity attached to a rule or fact expressing the certainty associated with it.

Chaining: The linkage of events which are caused by other events. There can be forward, backward, and mixed chaining.

Class: Many objects have common features, behaviors, etc. A class is merely a grouping or generalization of a set of objects. Objects are specific members or instances of a class.

Conflict resolution: The method an expert system utilizes to make a decision between conflicting facts. A particular object or class may often have several parents or even an entire network of possible parents from which to inherit. Each time an inheritance event occurs there might be a conflict between alternate sources of information. Usually inheritance strategies are utilized to resolve any conflicts.

Consequent: The right hand side of an IF\THEN rule.

Default value: A value used by the expert system if the actual value has not been specified.

Demon: A program, within a frame system, that is triggered when a particular action related to a slot occurs.

Development environment: The hardware and the software used during the development of an expert system.

Domain: The application area of an expert system.

Dynamic Link: A dynamic link is a runtime relationship between objects or dynamic objects and classes which is deleted at the end of the application run.

Dynamic Object: A dynamic object is an object that is created during the runtime execution of the application and is deleted at the end of the knowledge processing session. Dynamic objects inherit information such as properties and slot values from the class or object they are assigned to.

Expert: A person who, through training and experience, can perform a task with a degree of skill that is beneficial to capture and distribute.

Expert system: An advanced computer program that can, at a high level of competence, solve problems requiring the use of experience and expertise. Also, a sub-field of artificial intelligence related to the theory behind and the development and application of expert system programs.

Facet: A part of a slot that contains a piece of information related to the slot.

Fire: To activate or trigger a rule in a production rule system.

Forward chaining: Given a set of information and data, rules are applied and followed until a goal has been reached.

Frame: A structure containing information about a single item such as a concept, item, or class.

Heuristic: A rule of thumb or strategy that aids in solving problems or in making decisions.

Hypothesis: All rules have one and only one hypothesis, however, many different rules may lead to a hypothesis. A hypothesis is a boolean slot that will be evaluated to be true if all conditions of the rule evaluated to be true.

If Change: An if change specifies a list of changes that will be performed after the value of the slot is changed.

Inference engine: It is the part of the expert system which provides the system control.

Inheritance: A mechanism in a frame or object oriented system that allows all the information known in general about all the members of a class to be considered true for each individual member of the class, unless it is known to be different.

Knowledge acquisition: The process by which expert system developers find the knowledge that is used by domain experts to perform the task of interest.

Knowledge Base: The part of the expert system that stores the facts and heuristics of the domain expert. A knowledge base consists of objects, classes, properties, rules, and knowledge islands.

Knowledge Island: A knowledge island is a group of related rules. Rules within a knowledge island share hypotheses and\or information.

Knowledge implementation: The process of taking the knowledge found during the knowledge acquisition and translating it into an operational expert system program.

Knowledge representation: The process of defining the approach that will be used in an expert system program to represent the domain knowledge found during the knowledge acquisition.

Message: A communication sent from one object to the other.

Metaslot: Metaslots describe all aspects of the behavior of slots. All metaslots have default settings that control inheritance strategies, inference processes, order of sources, and if changes.

Mixed chaining: A reasoning technique used in production rule systems that allows both backward and forward chaining to be used for different parts of the same problem.

Multiple inheritance: Inheritance from more than one source.

Object: A data structure that contains all the information related to a particular item. An object is the smallest chunk of information in a knowledge based system. It represents any person, place, thing, or idea in the domain of a particular application.

<u>Object oriented programming:</u> A set of techniques that allows programs to be built using objects as the basic data items and actions on objects as the active mechanism.

<u>Pattern Matching:</u> Pattern matching allows the application to test the values of slots without having to mention them explicitly. Pattern matching creates a list of objects which belong to a parent class or object.

<u>Production rules:</u> The knowledge representation technique most often used in expert systems. Each rule represents one piece of knowledge and consists of a conditional and a prescriptive part.

<u>Property:</u> Properties have a particular data type. Properties can be a string, integer, float, boolean, date, or time. A property can also be defined to posses multi-values. Properties are usually assigned to a class or object . An object or class can have any number of properties associated with it.

<u>Rule based system:</u> A system based upon production rules.

<u>Slot:</u> A component of a frame that refers to a specific attribute of the frame entity and contains the value of the attribute if the value is known. The information may be

static or dynamic. Slots are used to store property values for objects and classes holding all the information of an application.

Subclass: A subclass is a class which represents a subset or specialization of another class. It is a class in its own right and has all the characteristics of other classes. Classes can have any number of subclasses or parent classes or both.

Subobject: Subobjects are not instances of each other but are also not completely distinct from each other. A subobject is also an object and can have a subobject itself.

Uncertainty: The situation in which knowledge or data in an expert system are not completely certain.

Working memory: The part of the expert system that contains the information the system has received about the particular problem at hand and any information that the expert system derives about the particular problem at hand.

APPENDIX A

EXISTING EXPERT SYSTEM APPLICATIONS

APPENDIX A

EXISTING EXPERT SYSTEM APPLICATIONS

A number of expert system applications have been recently, and are currently being developed in the area of structural mechanics (6, 7, 8, 9 , 10, 11, 12, 13, 14). Most of these applications address only one aspect of the analysis process, such as the control of the numerical solution algorithm, the preparation of the data input file for a specific program, advising on the finite element mesh creation and optimization, advising on best solution strategy, model consultation and verification, and the selection of the most appropriate finite element code to use in an analysis.

A few of the existing expert system applications in the field of structural mechanics extracted from (6, 7, 8, 9, 10, 11, 12, 13, 14) are discussed below in an attempt to show the versatility of this new technology and the directions in which current research is headed.

- ADEPT - ADEPT is an expert system that recommends a finite element analysis package most suited to a particular finite element problem. In addition, it makes recommendations on the type of finite element that should be used for a set of imposed boundary conditions and constraints.

- EXPERT - Expert is an expert system that assists the user in the selection of the optimum mesh design, monitors the process of the analysis, and provides guidance in the selection of the next appropriate processing step.

- FEASA - FEASA assists the user in the finite element specification and modeling. It also gives advice on how to select boundary conditions, design the mesh layout, and determine the most suitable mesh size.

- IQFEM - IQFEM uses expert system technology to help the user in the selection of an appropriate numerical algorithm for a nonlinear finite element analysis (i.e. the selection of the appropriate constitutive model, the type of iterative solution procedure that should be used, and in what way the convergence criterion should be selected and applied).

- SACON - SACON is an expert system that assists the user in the selection of the most applicable analysis strategy for a finite element analysis such as buckling analysis, nonlinear crack growth analysis, material instability investigation, etc..

- PRETAP - PRETAP is a pre - processor for a three - dimensional frame analysis program of tall buildings. It assists the user in the preparation, verification, and modification of the necessary information required by the analysis program. In addition, the program helps the user with the task of creating the data input file for the analysis program since it will perform all of the formatting of the input data.

- ETUDES - ETUDES is an expert system program that determines the optimum time step for the time integration of linear structural dynamic equations. In addition, the program determines whether an explicit or implicit method of solution will prove more efficient for a particular problem.

All of these applications prove to be valuable tools for the analyst, but it has been determined that not one expert system exists that combines all of the aspects involved in a nonlinear analysis into one complete package.

APPENDIX B

SELECTION OF EXPERT SYSTEM SOFTWARE

APPENDIX B

SELECTION OF EXPERT SYSTEM DEVELOPMENT SOFTWARE

INTRODUCTION

Expert systems are computer programs which attempt to simulate the intuitive portion of the human problem solving process. Expert systems (expert systems are a sub-branch included in the more general designation of knowledge based systems) are software programs written in a fourth generation computer language that use rules of thumb to solve complex problems through logic reasoning rather than the application of numerical algorithms. Typically, these programs are highly user friendly as a result of the simple to understand fourth generation programming language, and/or the extensive use of graphical interfaces. Similar to human experts, expert systems provide advice to the user by calling upon available knowledge stored in their working memory, or by asking the user for supplemental information needed to solve a problem intelligently and appropriately.

The "knowledge base" the expert system utilizes to arrive at a solution is usually stored as a series of

IF/THEN type rules in combination with objects, classes, and properties of the objects, specifically developed for a particular problem context. Supplemental information is supplied by the user in response to queries raised by the expert system. The "intelligence" of the expert system is derived from the use of efficient mechanisms which process and evaluate the information given by the user and the data and knowledge base in order to arrive at a solution or conclusion for a particular problem. This mechanism is called an "inference engine". So it may be stated, that there are two basic components which can be identified in almost all rule based expert systems:

1) The Knowledge Base
2) The Inference Engine.

The knowledge base is the collection of all of the known information in the form of rules. Each rule includes one or more conditional statements, and may include multiple conclusions. If the conditional statements are met, the expert system will assume the associated conclusion(s) as true and store them for subsequent use.

The inference engine is a computer program that examines the knowledge base and processes queries or responses of the user. In the inference engine, rules are "fired" in accordance with a set order established by the developer of the expert system. Inference engines can be classified as backward chaining or forward chaining. Backward chaining inference engines, which are goal driven, arrive at conclusions by evaluating what supporting conditions must be true to arrive at the specified goal. Forward chaining inference engines, which are rule driven, utilize some known initial conditions in order to determine the final solutions possible with the specified given information.

Within the subsequent sections, a total of five commercially available expert system software packages will be presented in detail, and their advantages and disadvantages will be discussed.

EVALUATION CRITERIA

The following evaluation criteria were used to determine the most adequate of the examined expert system software packages:

- The developers interface was evaluated on how easy it was for the user to develop an expert system application (i.e. how long does it take the novice user to learn all of the programming options contained in the software).

- The programs were judged on how their rule system can be defined and represented (i.e. the way the rules are created and verified by the application developer).

- Interest was paid to the methods that the program's inference engine employs to arrive at a solution (i.e. does the inference engine support forward chaining, backward chaining, or both forward and backward chaining).

- The development tools were evaluated based on their ability to represent the knowledge that is to be encoded (i.e. does the program support a rule system only, or a combination of a rule and object system).

- One of the more important aspects was the quality of the user interface that can be created with the development shell for the application (i.e. this was deemed important since the appearance of a program typically influences the level of confidence a user will have in an application).

- The graphics capabilities that the development shell possesses and the mathematical functions the program supports were evaluated (i.e. can graphic files be displayed or even used interactively within the application, mathematical functions are usually convenient and can reduce programming tasks).

- The programs were judged on their capabilities of being able to be integrated with other applications (i.e. programs can be integrated in modular fashion, or programs can run underneath other programs).

- Last, the development shells were evaluated on the quality of their user documentation, expert system application examples, and tutorials.

It should be noted that these evaluations of these software packages were done at the end of the year 1989 and may therefore at the time of this writing reflect old information.

EXSYS (3.2)

EXSYS is an expert system that uses IF/THEN/ELSE rules for the presentation of the knowledge base. Some of the more interesting features of this software package include the programming editor, the true forward chaining capability, and the ability to incorporate probabilities into the decision making algorithm. The EXSYS rule editor is more than just a text editor, it is a rule generation environment. Rather than typing in the rules using the programming language, the user is presented with options within the rule development. Each item or condition need only to be introduced once, and may be recalled at any time during the development phase. This will result in a more structured development of the program, and in less typographical errors within the program application code. Each rule which is entered is also checked against all other existing rules, and any conflicts between the new rule and the existing rules will be flagged and displayed immediately to the user for correction.

EXSYS has the capability of forward and backward chaining in its inference engine, and is even capable of a mixed mode of chaining, which can be very useful at times. In order to maximize the speed of execution of the developed expert system, EXSYS will rearrange the rules to optimize the performance. EXSYS is able to execute knowledge bases with up to 5000 rules on any PC computer, and virtually an unlimited number of rules on a VAX/VMS system. Another feature of EXSYS is, that it allows linking of various developed applications to a control expert system, which will run the dependent expert systems depending on need. It is therefore possible to separate a large complex problem into components which are more manageable, therefore reducing the time necessary to develop the expert system application.

The user is able to choose among three different types of logical schemes, a true Boolean approach, a probability based approach, and a subjective evaluation of confidence expressed in a range of numerical values. Probabilities can be averaged, minimized, or maximized to arrive at a solution. The sensitivity of the solution can be determined by changing some of the parameters or the input for the expert system. The program will display information about how the changes in the input affected the previous conclusion, and how it arrived at the new conclusion.

EXSYS allows the presentation of graphics developed from other software packages. It is also able to manipulate other programs in order to revise graphical information depending on changes in the knowledge base or other information. External programs can be called and activated from within the program and data can be exchanged and updated. EXSYS supports a variety of mathematical functions within its rule based code such as trigonometric functions, logarithmic and exponential functions, etc.

EXSYS comes with a variety of documentation, including a manual, a tutorial, and small example expert systems. The tutorials and the example expert systems should be expanded in order to train in some of the more advanced features of the software program. The manual of the program was difficult to follow and was not organized well.

1ST CLASS

1ST CLASS uses the spreadsheet approach to the development of expert systems. The program contains an extensive rule editing system in which rules are made and executed. The program, composed of a series of six menu driven screens in which all of the commands are located, creates much confusion, especially since the user is required to backtrack through all intermediate screens if he/she wants to move from the sixth to the first screen.

Rules can not be written directly in 1ST CLASS, but rather have to be developed by example. This results in decision trees developed in a spreadsheet format. These rules are composed of root nodes, branches, paths, and result nodes. This system can be much more difficult to understand than a system of linguistic rules, especially for the novice.

Knowledge is represented as either numeric or work factors. The maximum length for any variable is eleven characters, which limits the user severely in the development of larger expert systems. Both forward and backward chaining are available in the 1ST CLASS software package. In addition, two other approaches are available to evaluate the rule base. Knowledge bases can be chained together, which proves beneficial, since a single system is limited to either 32 factors or results. Some graphics are available in 1ST CLASS for the development of decision trees, but any other graphics have to be generated by supplementary programs able to interface with the development tool.

The written documentation for 1ST CLASS is marginal, since the material is not presented in a logical order, and information on a single topic is usually spread throughout the user's manual. The tutorials included with the program are not easy to understand, which further contributes to the difficulties associated with this particular development package. This particular expert system software program is just not well suited for the use in a larger application development.

GURU (1.1)

GURU is a truly integrated software package that contains a word processor, database manager, spreadsheet, communications package, natural language interface, and expert system. The integration of these modules performs well, allowing the user to transfer information and commands between systems in an organized, consistent fashion.

GURU offers two developer interfaces, a special rule writer and a standard text editor. The rule editor divides the screen into a series of windows, one associated with each portion of the expert system format. In addition, when editing specific rules, GURU divides the screen into windows for each rule function. This provides a useful structure for rule development for novices.

Several forms of inference mechanisms are available to the user, including forward and backward chaining, with and without logic tracing, and drivers for the inference engines contained in the rules which can set priorities for the firing of rules. As with most expert systems, backward chaining is the default inference procedure, but the user can specify any of the other available inference methods. GURU allows the user to specify precisely the order in which rules are fired and the logic used for selecting subsequent rules to be added on the agenda. External programs can be executed when a rule is fired, but the interfacing is somewhat limited.

The user interface of GURU works well, because the interface with the user is controlled in the description of rules. This allows the user to present questions and information in the best possible manner without extensive efforts. In addition, because GURU is an integrated package, graphic capabilities, database management, and spreadsheet capabilities can always be utilized to aid in the clear presentation of information. At the same time, since spreadsheets can always be called from the expert system, mathematical functions are essentially only limited by the user's knowledge of the spreadsheet's capabilities.

GURU (written in C programming language) contains many helpful tutorials. The GURU tutor, which is a feature of the system, enables the user to learn quickly about all of the system's fundamental capabilities within a four to six hour session. Example knowledge bases are included to demonstrate the use of GURU in a wide variety of settings and for a variety of purposes. GURU is one of the most complex expert systems on the market, and the integration features suggest that a long training period is usually necessary to become familiar with the program.

LEVEL-5 OBJECT (2.2)

In keeping with the increasing requirements for object oriented programming in today's computing environment, LEVEL-5 is completely object oriented. The editors, visual windowing system, displays, database interfaces, inference engines, knowledge bases, devices, files, and timers are all objects built into the LEVEL-5 program. LEVEL-5 packages these objects as systems classes that contain an array of built in logic and object tools, which give the developer ultimate control the application design and function. These built in objects are very useful templates for the user who wants to create new object classes. The default definitions serve as a starting point, which will reduce the time required to develop a customized application.

LEVEL-5 supports multiple inheritance of the knowledge base, since the logic and data structures expressed as object classes can be reused, with the newly created objects inheriting their attributes from one or more parent objects. The inference mechanism controls how LEVEL-5 pursues stated goals, applies rules,performs queries, uses the objects and methods, and reaches conclusions. The inference mechanism can process information by using backward chaining, forward chaining, parallel processing, creating dynamic agendas, applying rules, and object oriented programming. The versatility of the inference mechanism in the LEVEL-5 program enables the user to represent the knowledge fully customized to the needs of the expert system application.

The user interface of the LEVEL-5 program is invoked through Microsoft Windows. The graphical user interface takes full advantage of the desktop metaphor since it utilizes pull down menus, dialogue boxes, check boxes, radio buttons, and more. With the help of the graphical toolbox, developers can build these same display facilities into their expert system applications. The rule talk module, which is an interactive facility, maintains a tight integration of all the other software modules, rules, objects, declarations, knowledge trees, and displays.

LEVEL-5 runs on an extensive range of operating systems, including the IBM PC, the Apple Macintosh, the VAX\VMS, and even the IBM VMS and MVS mainframe platforms. LEVEL-5 expert system applications can therefore be developed within the IBM PC environment, and at a later date be ported to a mainframe.

The program has extensive math functions capabilities, such as trigonometric and logarithmic expressions. Once a knowledge base has been entered, it has to be compiled. Since the knowledge base will be compiled, the program is able to execute faster, and compiled systems can be distributed at a much lower cost to other users since they comprise runtime only versions rather than full development versions. While LEVEL-5 does not have the facility to display graphics in its own structure and code, other than graphics based on the Microsoft Windows environment, it can be instructed to display screens created by a variety of popular graphics manager software packages. Consequently, drawings and illustrations can be easily incorporated into the knowledge base, making it possible for the expert system developer to enhance the user interface.

The graphical interface of LEVEL-5 using the Microsoft Windows makes the program intuitive to use even for the inexperienced expert system application developer. Nevertheless, the developer will need to understand the concepts and methodologies of the expert system methodology well in order to develop an expert system application. However, LEVEL-5 comes with excellent documentation and a brief tutorial which will enable the user to familiarize himself quickly with the program. LEVEL-5's debugging aids are another help for the inexperienced expert system developer. Using a function key labeled "why?", the user can explore the logic chain followed by the software as it reached its conclusions. With similar facilities, the user can explore the facts developed in a session.

During the development phase of an expert system project, it is useful to be able to manipulate the conclusions of rules or the input of the user. LEVEL-5 allows the user to change variable contents and to observe the effects of single changes to the input without re-entering all data.

While LEVEL-5 is not the easiest expert system to use, it has excellent documentation and debugging aids, which will enable the inexperienced user to develop small expert system applications on his/her own within a relatively short time period.

NEXPERT OBJECT (2.0)

Written in the C programming language, NEXPERT is a rule and object based expert system software tool featuring a powerful user interface, open architecture, and unique reasoning capabilities. The graphical user interface makes using NEXPERT intuitive even for those with little or no background in expert system development. It's open architecture allows for easy integration with conventional software such as C, COBOL, PASCAL, FORTRAN, and Compiled BASIC.

NEXPERT uses a graphical interface technology, which is called "Open Interface Toolbox" (OIT), which is a completely portable software development environment for graphically creating user interfaces across all platforms. OIT allows interfaces developed on one machine to be ported without modification to other machines. For example, an application created under Microsoft Windows could be moved without modification or reprogramming to a SUN workstation with X-Windows, or to a Macintosh computer under Apple's proprietary windowing system.

The development of an expert system application using NEXPERT is said to be intuitive and easy. NEXPERT's hybrid design combines rules for inference and control with objects for representing things and ideas. A highly graphical interface provides editors for rules, objects, classes, properties, metaslots, slots, if changes, and other NEXPERT elements. NEXPERT's editing environment makes extensive use of pop up menus to help simplify entry and save keystrokes associated with commands. The NEXPERT program will check the syntax of rules right after they have been entered and will immediately flag any inconsistencies with the already existing rule base. Once the rule is entered in the editor, it will become an active member of the current knowledge base. Objects, classes, properties, and other elements are also compiled in increments. The editors make NEXPERT easy to use, but it is the inference engine that gives NEXPERT its reasoning power.

NEXPERT's primary inference methodology is called opportunistic reasoning. In most expert systems, developers must choose between a forward- or backward-chaining inference engine. Backward-chaining systems begin their inference with a conclusion, then proceed backward through the knowledge base, trying to confirm or deny the statement. Forward chaining systems take data as input and then search for rules that use this data and try to determine what conclusion to draw. NEXPERT combines these two methods, so that the developer of the expert system application can create an expert system that more precisely emulates the human expert.

In addition to the above mentioned characteristics of the NEXPERT software package, the program facilitates all of the characteristics of the EXSYS and LEVEL-5 software packages.

RECOMMENDATION

During the course of the research project it was recommended, that BPA acquire the NEXPERT OBJECT expert system development shell in order to provide PSU with the adequate means to develop the Limit States Analysis Module application. It was felt that NEXPERT OBJECT was the best choice since it allowed the LIMIT ES program to run in sub-modular fashion because MS Windows is not required for the runtime executables. Superior quality of the above recommended software package as indicated in Table I: "Summary of Review of Expert System Software Packages" warranted the use of it in this particular application development.

All of the information was taken from available product literature of the software development tools. Judgements and preferences are based on available product literature of the software programs and experience gained from promotional instructional seminars which are frequently offered by the software developers. The rankings given to the program options as shown in the "Review Summary" are somewhat subjective, since the tools were evaluated in regard to a specific purpose (i.e. the development of the Limit States Analysis Module). A different problem context could have changed the ranking assigned to certain program options and capabilities.

Many other expert system development tools are currently available, which the author has selected to omit from this section. The decision to exclude these other development tools from the review process presented in this section has been made with regard to the specific problem context of the Limit States Analysis Module research and development. In addition, it may be stated, that a complete review of all the commercially available expert system development tools is not within the scope of the research and development presented in this document.

TABLE I

SUMMARY OF REVIEW OF EXPERT SYSTEM SOFTWARE PACKAGES

	EXSYS	LEVEL-5	NEXPERT	GURU	1ST CLASS
Developer Interface	5	5	5	5	3
Rule System	4	4	5	5	3
Inference Engine	3	5	5	5	3
Knowledge Representation	3	4	5	4	3
Uncertainty Management	4	5	5	4	3
User Interface	4	5	5	4	3
Graphic Capabilities	3	4	5	4	3
Math Libraries	4	4	4	5	2
Integration Capabilities	4	4	4	4	3
Language & Execution Speed	4	5	5	5	3
Documentation	3	5	5	4	3
Tutorials	4	5	5	5	2
Price Effectiveness	4	4	4	4	3
Subjective Evaluation	4	5	5	5	3

Ranking: 5 - Excellent

4 - Very Good

3 - Good

2 - Marginal

1 - Poor

APPENDIX C

DESCRIPTION OF THE LIMIT STATES ANALYSIS MODULE

APPENDIX C

DESCRIPTION OF THE LIMIT STATES ANALYSIS MODULE

COMPUTER SYSTEM CONFIGURATION

The Limit States Analysis Module can currently be used on any IBM PC compatible 80386 or 80486 computer system that has a hard-disk and a VGA graphics display. The program requires that a numerical co-processor is installed and it is recommended that the system has at least 4 megabytes of RAM. The program is run in the DOS operating system environment. It is recommended to use MICROSOFT DOS Version 5.0.

The Limit States Analysis Module can be installed from the floppy drive to the hard-disk by changing to the floppy drive directory and typing "INSTALL". The installation routine will prompt the user for information on what logical drive the user wants to install the program to. The installation routine will create the necessary directory structure and copy the appropriate files to the hard-disk. Upon completion of the file transfer the installation routine will end and return the user to the previously defined hard-disk drive. A complete directory structure and listing of all the required files to run the Limit States

Analysis Module can be found in Appendix D.

DESCRIPTION OF PROGRAM MODULES

A number of program modules have been developed that work together to form the Limit States Analysis Module. A variety of programming languages were used to develop these components. The NEXPERT OBJECT development shell has been used to develop the expert input and output evaluations, MICROSOFT PROBASIC Version 7.1 was used to develop the user interfaces and graphical plotting routines, and the LIMIT B program was changed to use the SALFORD FORTRAN 77 compiler. In addition, other programs that are commercially available have been incorporated to serve as a text editor and to display graphics based help files. A description of the more important program modules is presented on the following pages:

EXPIN.KB - EXPIN.KB is the file that holds the developed and compiled expert input evaluation knowledge base. The input evaluation module was developed with the NEXPERT OBJECT development shell.

EXPOUT.KB - EXPOUT.KB is the file that holds the developed and compiled expert output evaluation knowledge base. The output evaluation module was developed with the NEXPERT OBJECT development shell.

EXIN.RTD & EXOUT.RTD - EXIN.RTD and EXOUT.RTD are the runtime definition files that activate the inferencing process for the expert input and output evaluation respectively. The runtime definition files were developed using a standard text editor.

TEMPLTE1.TXT & TEMPLTE2.TXT - TEMPLTE1.TXT and TEMPLTE2.TXT are format files that define the appearances of the expert input and output evaluation respectively.

NXPFORMS.EXE - NXPFORMS.EXE and all of the other files that have the prefix "NXP" are files that are necessary to run the expert system evaluations as stand-alone applications without using the development shell.

LIMIT4.EXE - LIMIT4.EXE is the executable nonlinear analysis file written and compiled using the SALFORD FORTRAN 77 programming language.

START.EXE - START.EXE is the executable file written in BASIC 7.1 that starts the Limit States Analysis Module.

CONTROL.EXE - CONTROL.EXE is the module that keeps track of the progress of the Limit States Analysis Module analysis and determines what options are available to the user.

CONVERT.EXE - CONVERT.EXE translates all relevant input and output information into a format that can be accessed by the NEXPERT OBJECT database interface.

TOWER.EXE - TOWER.EXE is the program module that allows the user to graphically plot the structural model and the post-buckling member performance behavior curves.

LOADHIST.EXE - LOADHIST.EXE plots the load history of user selected members after the completion of a successful nonlinear analysis.

DBOS.EXE - DBOS.EXE and all other files with the prefix "DBOS" are proprietary runtime and library files that allow the LIMIT ES analysis program to run using the SALFORD FORTRAN 77 programming language.

MAINMENU.EXE - MAINMENU.EXE is the program module that contains all of the user menus, screen editors, and read\write source code of the Limit States Analysis

Module.

CURVE.DAT - CURVE.DAT is a data file that contains all the information about the post-buckling member performance model curves or actual curves that are used by the LIMIT ES analysis program.

There are many other files that make up the Limit States Analysis Module. Many of them are files that pertain to the on-line help screens, graphical help displays, the on-line LIMIT ES user and modeling manual, etc. A complete listing of all of the files required to run the Limit States Analysis Module and the directory structure is given in the Appendix.

USING THE LIMIT STATES ANALYSIS MODULE

Startup

The Limit States Analysis Module is started by changing to the C:\LIMIT directory and typing "START". The program will begin by displaying the Limit States Analysis Module logo followed by subsequent screens that display credits to the developers and sponsors and a standard disclaimer. Screen shots of the credits and the disclaimer can be seen in Figures 29 and 30 respectively. After the user chooses to continue, the program will proceed to the Limit States Analysis Module main menu.

Main Menu

A screen shot of the Limit States Analysis Module main menu can be seen in Figure 31. The screen displays the name of the menu, the date, and the time of day. There are four major sections that are displayed on the main menu screen. The sections are the file, input, analysis, and output options that are available to the user. Options that are available to the user are highlighted by a first letter colored white.

As can be seen in Figure 32, the user at the beginning of the Limit States Analysis Module has only certain options available. At the start of the program the user is only able to use the filename, editor, and user and modeling manual options. The user is able to activate any of these options by entering manually on the keyboard the first letter (i.e. the letter that is highlighted) of the available option.

The program uses two features to guide the novice user through an analysis. The first feature is the command line that will indicate to the user what function key options are available to him\her. The command line is always the second line from the bottom of the screen. In Figure 31 the command line displays two function key options; one that will activate on-line help, and the other that will exit the program. In addition, the program has a prompt line that indicates to the user what the next logical step in the analysis is and how he\she should proceed. The message line, located on the very bottom line of the screen, shown in Figure 31 suggests to the user to enter the letter "F" which will activate the filename option.

Once the user chooses "F", the screen will change to the input directory which can be seen in Figure 32. The input directory will be discussed in detail in the following section.

Input Directory

The input directory has a similar appearance as the main menu. Again, the name of the menu, and date and time are displayed at the top of the screen. The overall layout of the input directory is shown in Figure 32. A blinking cursor prompts the user to enter any one of the file names that are displayed in the window below. LIMIT ES data input files can be recognized by their ".LM1" file extension and are always located in the C:\LIMIT\I&OFILES subdirectory.

If the file that the user wishes to work with is not shown in the display window, he\she may choose to shell to the MS DOS operating system in order to copy the appropriate file into the subdirectory. Typing "EXIT" from anywhere within the MS DOS operating system will return the user to the Limit States Analysis Module.

Once a filename has been entered, the input directory screen will disappear and the output directory will be displayed. A screen shot of the output directory can seen in Figure 33 and is discussed in the following section.

Output Directory

The main feature of the output directory is the display window that shows all of the output file names that can be created in an analysis using the Limit States Analysis Module. Figure 33 displays the filenames of all the output files that can be created. The output filenames only differ in their filename extension and a short description of their content is displayed in the window. Files that exist at this stage of the analysis will be marked by an arrow. It can be seen in Figure 33 that a LIMIT ES analysis was previously run for this particular data input file since a load history and output file already exist.

Once the user chooses to continue the output directory will disappear and he\she will be returned to the main menu screen to proceed with the analysis or to make changes.

View\Print File

Upon returning to the main menu, the user might want to choose to view or print either the analysis output or the pre-processor file. To perform this task the user would choose the view\print file option on the main menu. Once the option is activated, the main menu will disappear and will be replaced by the view\print menu shown in Figure 34. At this point the user will be able to view on the screen, or create a hard copy of, any of the files that are displayed within the window and are marked by an arrow. If the view option is activated the view\print file menu will disappear and the screen will change to the limit file viewer displaying the selected file. An example of the limit file viewer screen displaying the pre-processor report is shown in Figure 35. All options that are available to the user are shown on the bottom line of the screen, the command line. Once the user chooses to exit the limit file viewer he\she will be returned to the view\print menu. The user is now able to either create a hard copy of any of the existing files or to exit the menu. If the user chooses to exit the view\print menu he\she will be returned to the main menu.

Input Menu

As soon as a valid Limit States Analysis Module data input file has been specified the user will be able to activate the input menu. There are two options that are available to the user once the input menu has been activated.

The second option activates NEXPERT OBJECT's inference engine and performs the expert input evaluation. Once the inference process is completed the evaluation report is displayed on the screen and can be viewed by the user. If the user chooses to continue the expert input evaluation is saved in the C:\LIMIT\I&OFILES subdirectory and the user will be returned to the main menu.

The user can also first activate the input option and then the parameter option displayed in the main menu. Once the parameter option has been activated, the main menu screen will disappear and the input menu screen is displayed. Figure 36 shows the input menu with all the options available to the user. From within this menu the user is able to change any of the values contained in the LIMIT ES data input file. Screen-shots of the various screen editors available to the user are shown in Figures 37 through 43.

Figure 43 shows a picture of the bandwidth optimization menu that allows the user to minimize the bandwidth of the structural stiffness matrix. Reducing the bandwidth of the structural stiffness matrix reduces the computational time required to analyze a tower. There are two options available to the user in the menu. Activating the first option, the manual optimization process, allows the user to calculate the bandwidth with respect to one specific joint number. The user can then choose to enter a different joint number and the process will repeat. If the second option is activated, the automatic bandwidth optimization, the bandwidth will be calculated for each joint of the structural model. Eventually, after the bandwidth has been calculated for each joint, the program will display the optimum seed joint and the corresponding minimum bandwidth of the structural stiffness matrix. Upon exiting of bandwidth optimization menu the user will be asked if he\she wants to change the seed joint number to the number of the joint that was calculated to be the optimum seed joint.

Output Menu

There are three options that can be activated after the output menu has been activated from within the main menu. The history graphics option can only be activated if a LIMIT ES output file and a load history file exist for the specified data input file. The graphic module including its curveplot subroutine can be accessed once a LIMIT ES checkrun or full analysis has been run. The expert output evaluation requires that a full LIMIT ES analysis was performed.

Once the history graphics option has been activated the main menu screen will disappear. The user will be asked which members's load history he wishes to view. A total of four members can be displayed at any one time in the history graphics module. As soon as the selected member labels are entered and the user chooses to continue, the screen will change to display the load history of the members. A typical history graphics screen shot is shown in Figure 44 and is based on results of a LIMIT ES analysis using bilinear post-buckling member performance behavior. Two of the members in Figure 44 are stressed in tension while the other two are stressed in compression. A visible change of slope can be determined at a load factor of about 1.4 which indicates that these members started to perform inelastically at that point of the loading stage. The labels of the members that were selected for the display are listed at the top of the screen in ascending order from left to right. At this point, the user has the choice to either select a different set of members to be displayed or to exit the history graphics module and to return to the main menu.

The second option available to the user in the output menu is the graphics module that contains the tower plotting routine and the curveplot post-buckling member performance display. Once the graphics module has been activated the towerplot-curveplot opening screen will replace the main menu screen. An example of the opening screen is shown in Figure 45 containing the current file name and some other summary information that is helpful to the user.

If the user chooses to continue the opening screen will be replaced by the towerplot display. An example of the towerplot routine is shown in Figure 46 with its default settings. All of the options that are available to the user are displayed on the right hand side of the screen. Options can be activated using the appropriate function keys. If the user chooses to activate the curveplot option, the towerplot screen will disappear and the curveplot graphics routine will appear. Figure 47 shows an example of how the curveplot graphics routine would appear on the computer screen. The display captured in Figure 47 shows the bilinear post-buckling member performance model curve that was assigned to the member. The thin vertical line shows at what point the member is performing. Through the position of the vertical line one can see that the member reached its compression capacity and has yielded subsequently. The nonlinear curve displayed in Figure 47 shows the true, experimentally obtained, nonlinear post-buckling member performance behavior curve for a compression member with the geometric and strength characteristics specified in the LIMIT ES data input file. Note, that both, the bilinear model performance curve and the true performance curve display the same compression capacity but not the same peak displacement. Even though the load shift has been performed automatically by the program the user still has to shift the true performance curve. Information on how to shift the experimental curve can be accessed by the user through the on-line help or by referencing to the on-line modeling manual. If the user chooses to continue he\she will be returned to the towerplot graphics routine.

The third option available in the output menu is the expert output evaluation. Once the output evaluation is activated, NEXPERT OBJECT's inference engine takes control and evaluates all the rules in the knowledge base exhaustively until all hypotheses have been checked. The output evaluation is then displayed on screen until the user chooses to continue and to return to the main menu. The expert output evaluation file will be saved automatically to the C:\LIMIT\I&OFILES subdirectory where it can be viewed or printed later.

LIMIT STATES EXPERT SYSTEM

CIVIL ENGINEERING
SCHOOL OF ENGINEERING AND APPLIED SCIENCE
PORTLAND STATE UNIVERSITY

W. H. MUELLER
M. OSTENDORP
D. M. HESSE
J. A. ROBINSON
S. C. SJURSET

RESEARCH SPONSORED BY
BONNEVILLE POWER ADMINISTRATION

L. KEMPNER JR.
M. D. MILLER

Figure 29. Limit States Analysis Module Credit Screen.

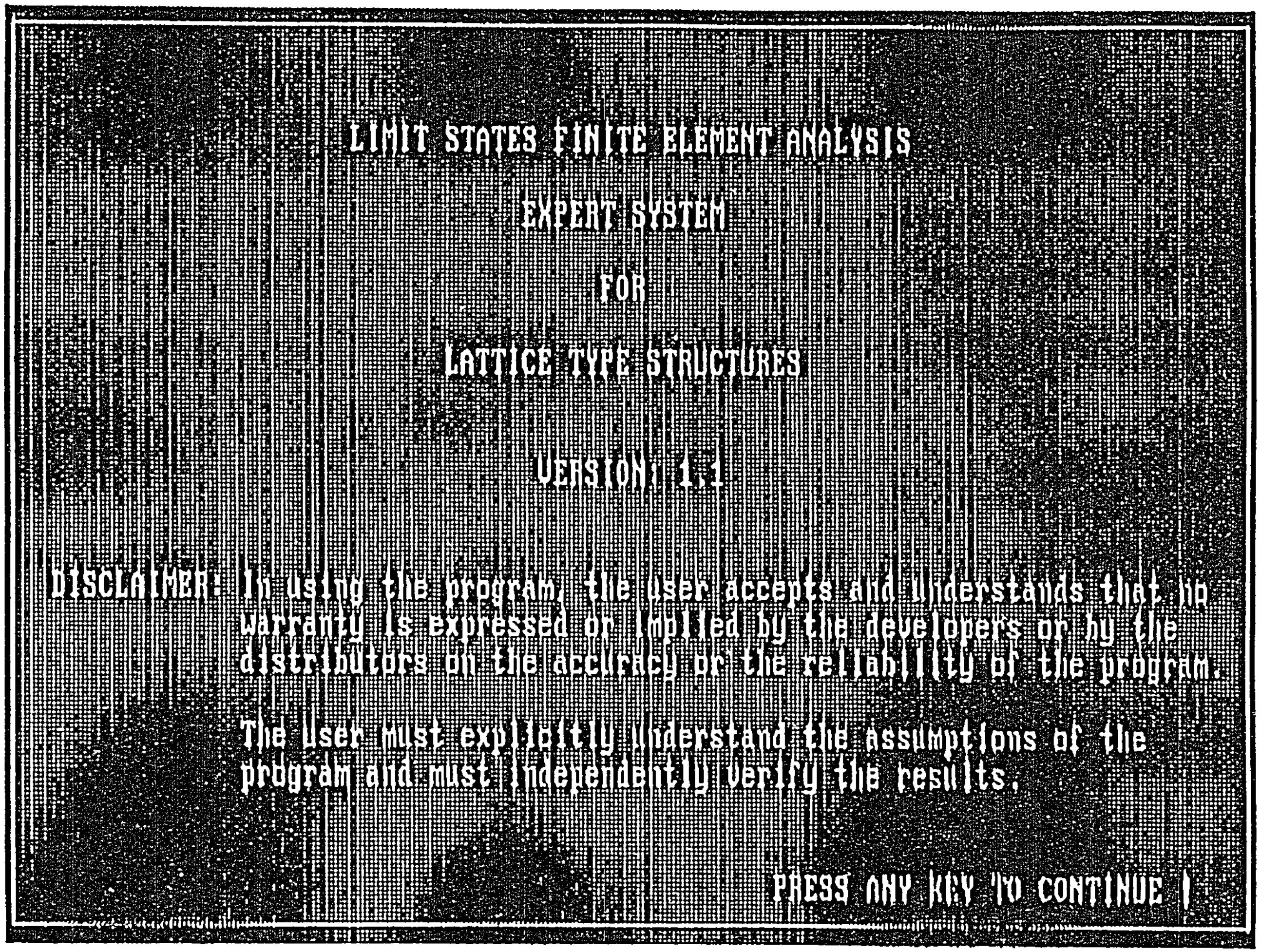


Figure 30. Limit States Analysis Module Disclaimer.

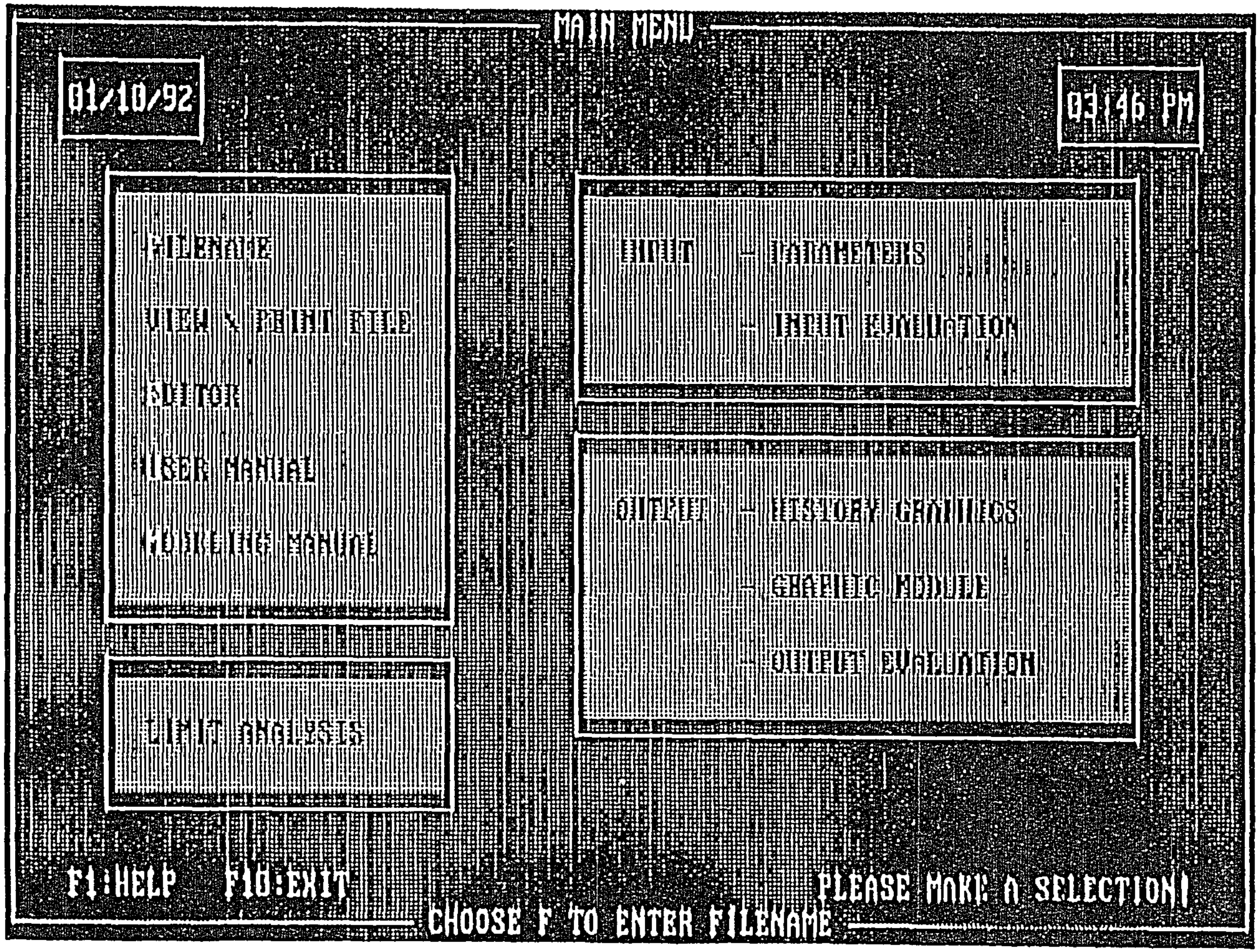


Figure 31. Limit States Analysis Module Main Menu.

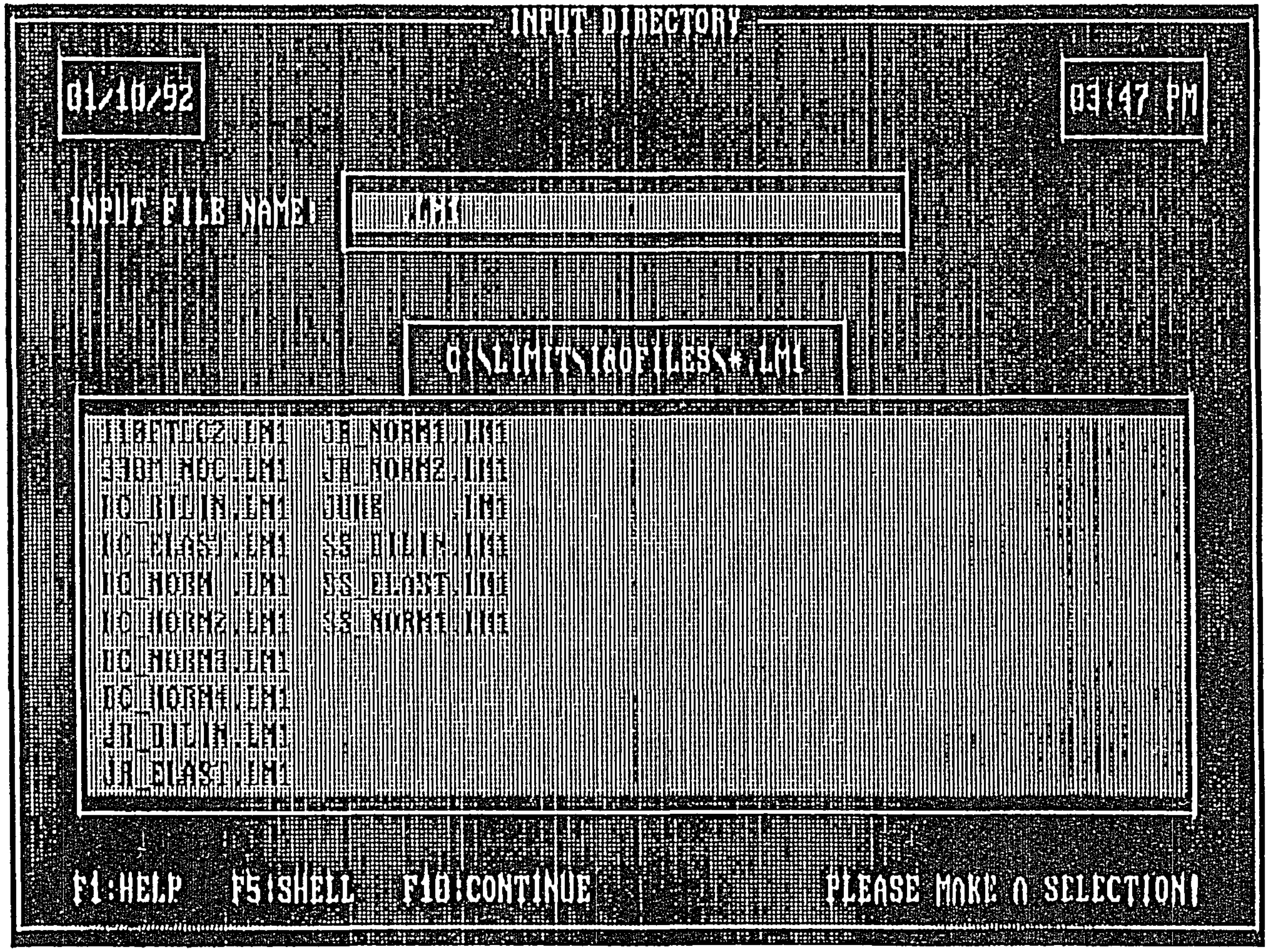


Figure 32. Limit States Analysis Module Input Directory.

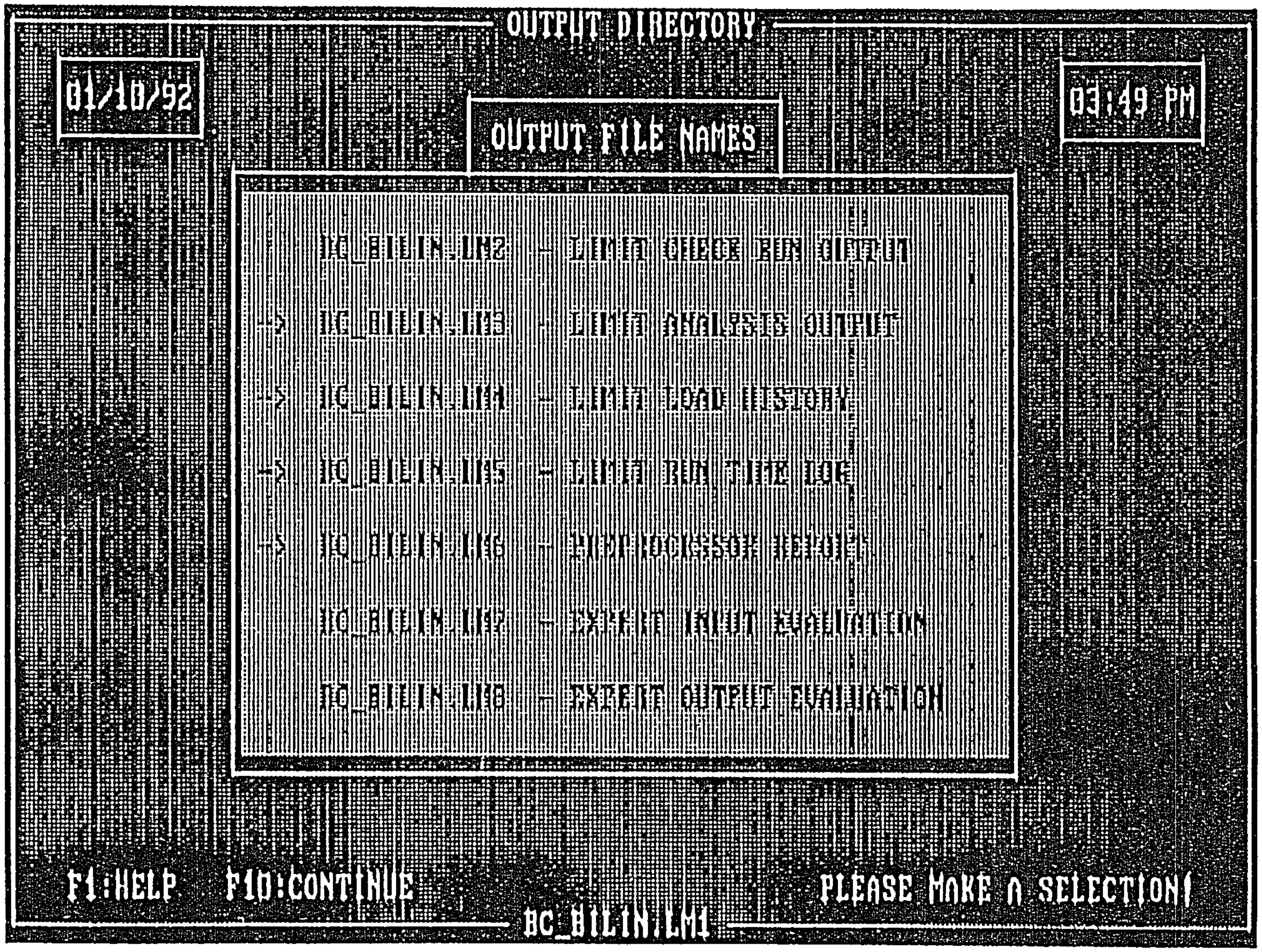


Figure 33. Limit States Analysis Module Output Directory.

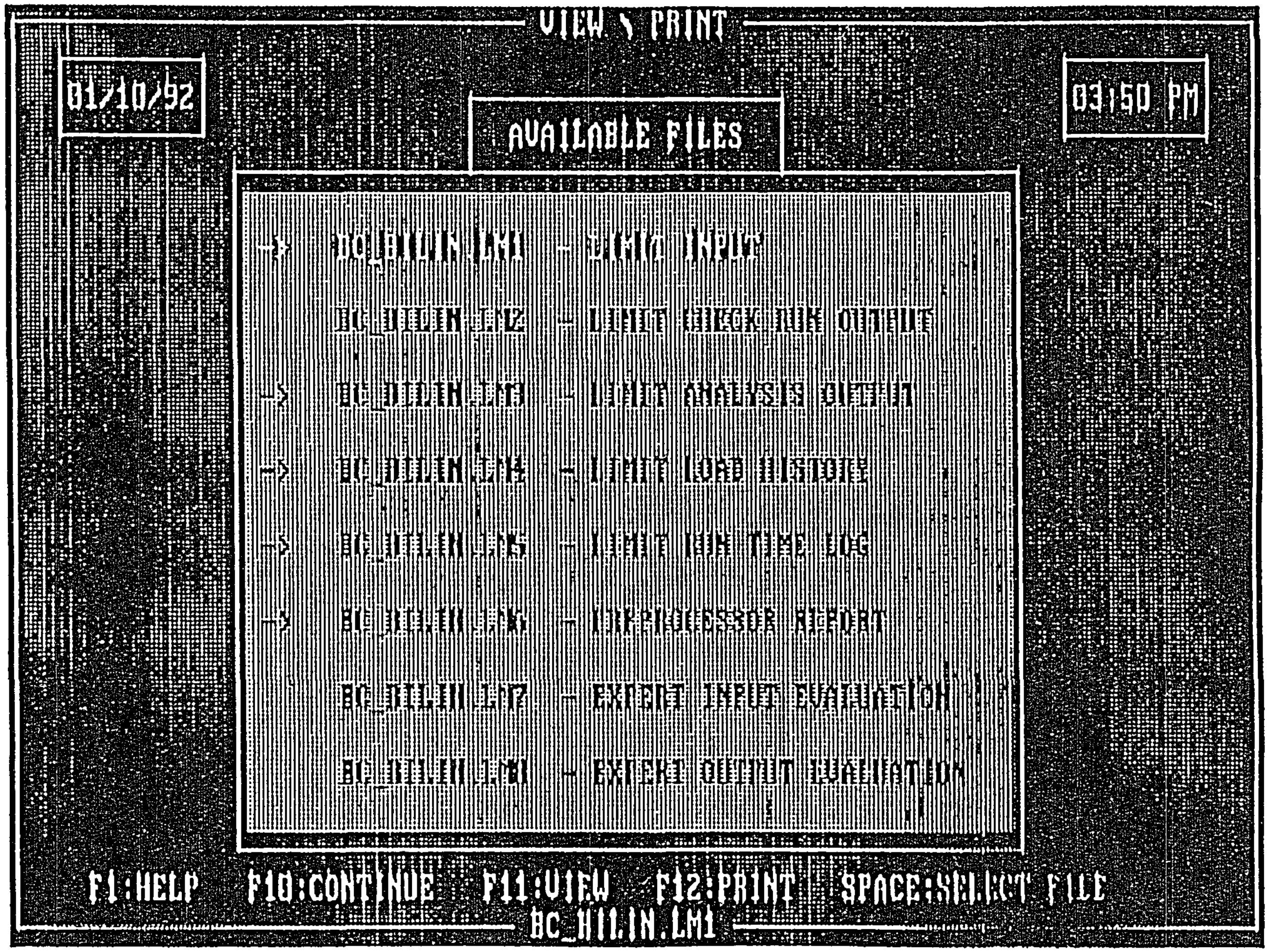


Figure 34. Limit States Analysis Module View\Print File.

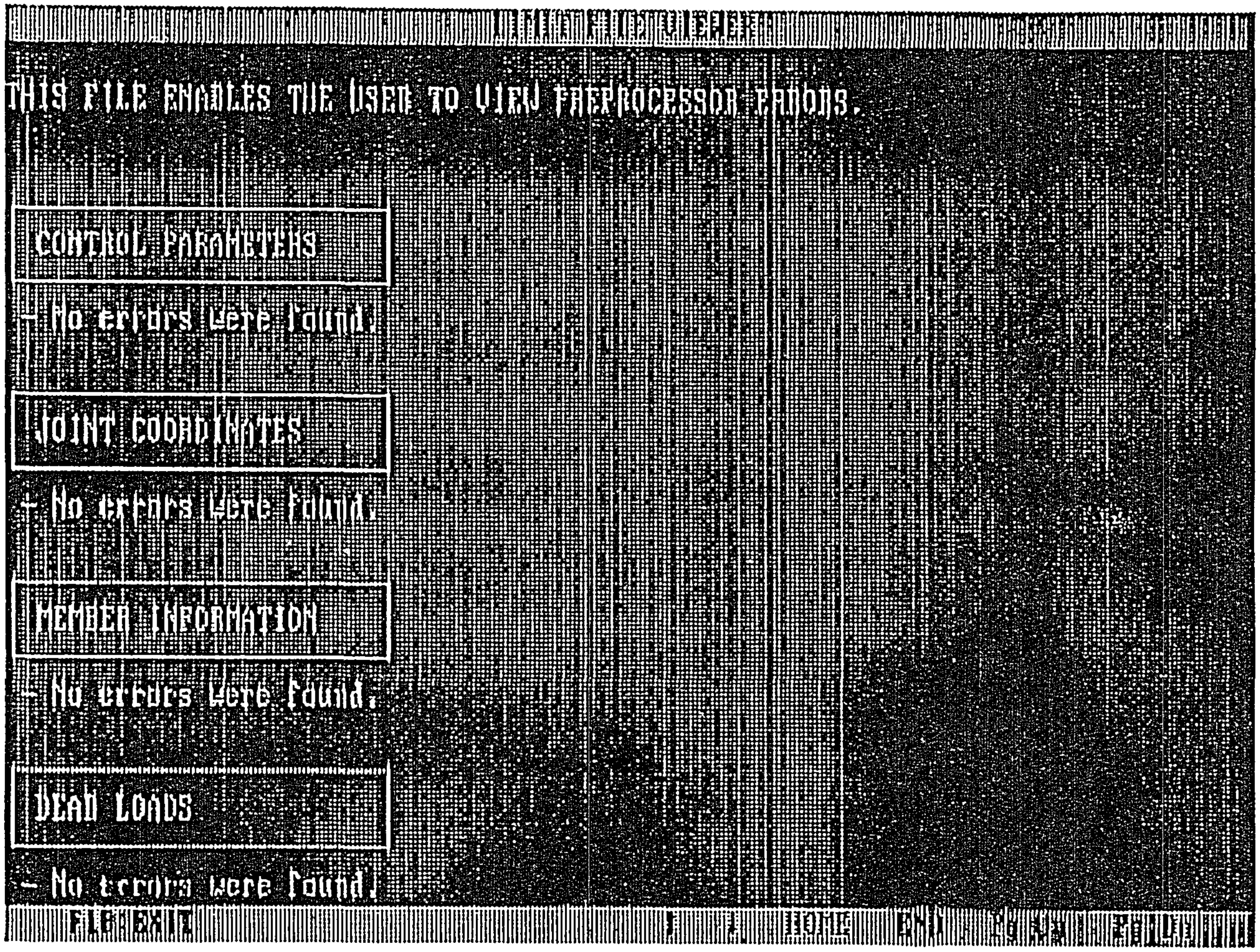


Figure 35. Limit States Analysis Module Limit File Viewer.

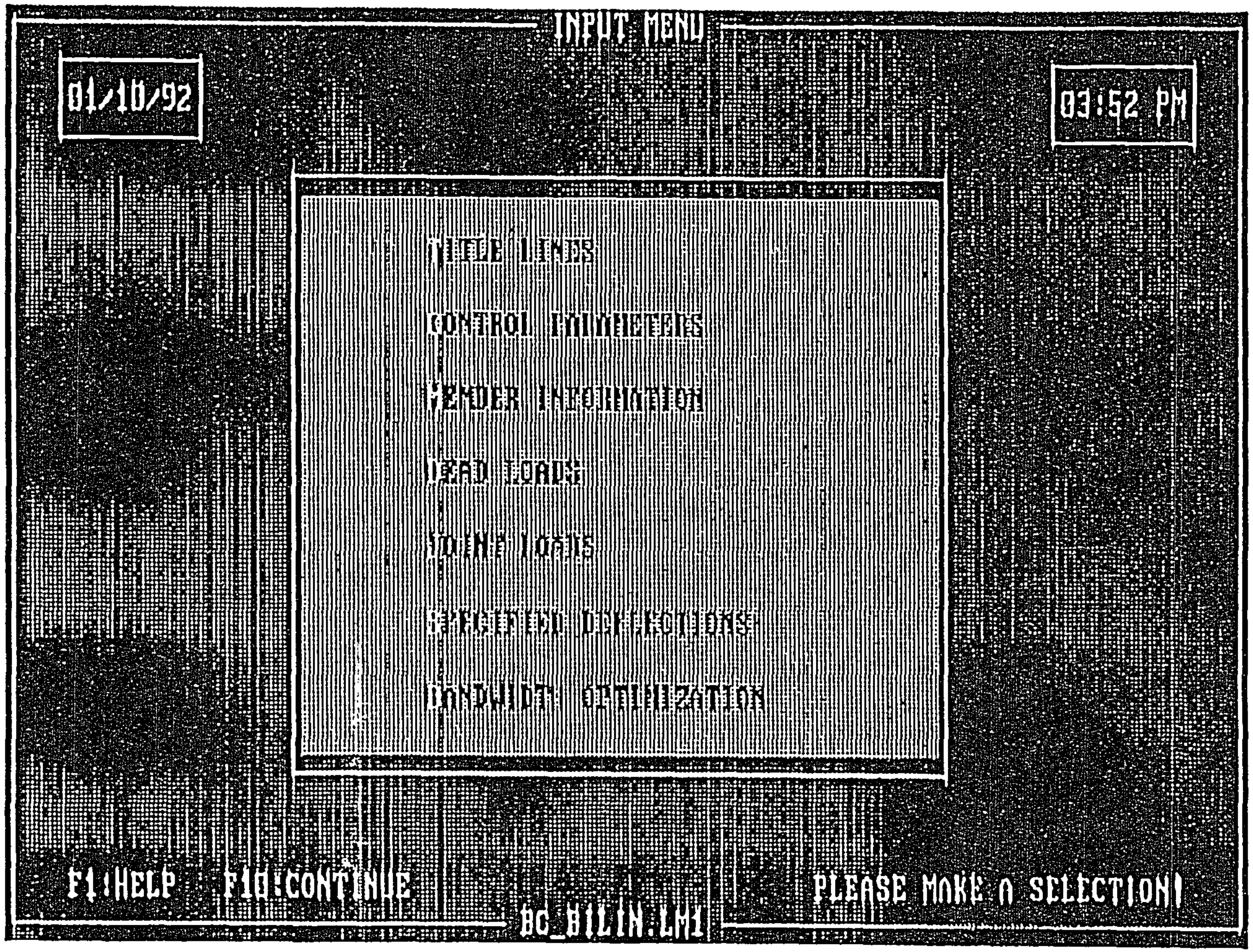


Figure 36. Limit States Analysis Module Input Menu.

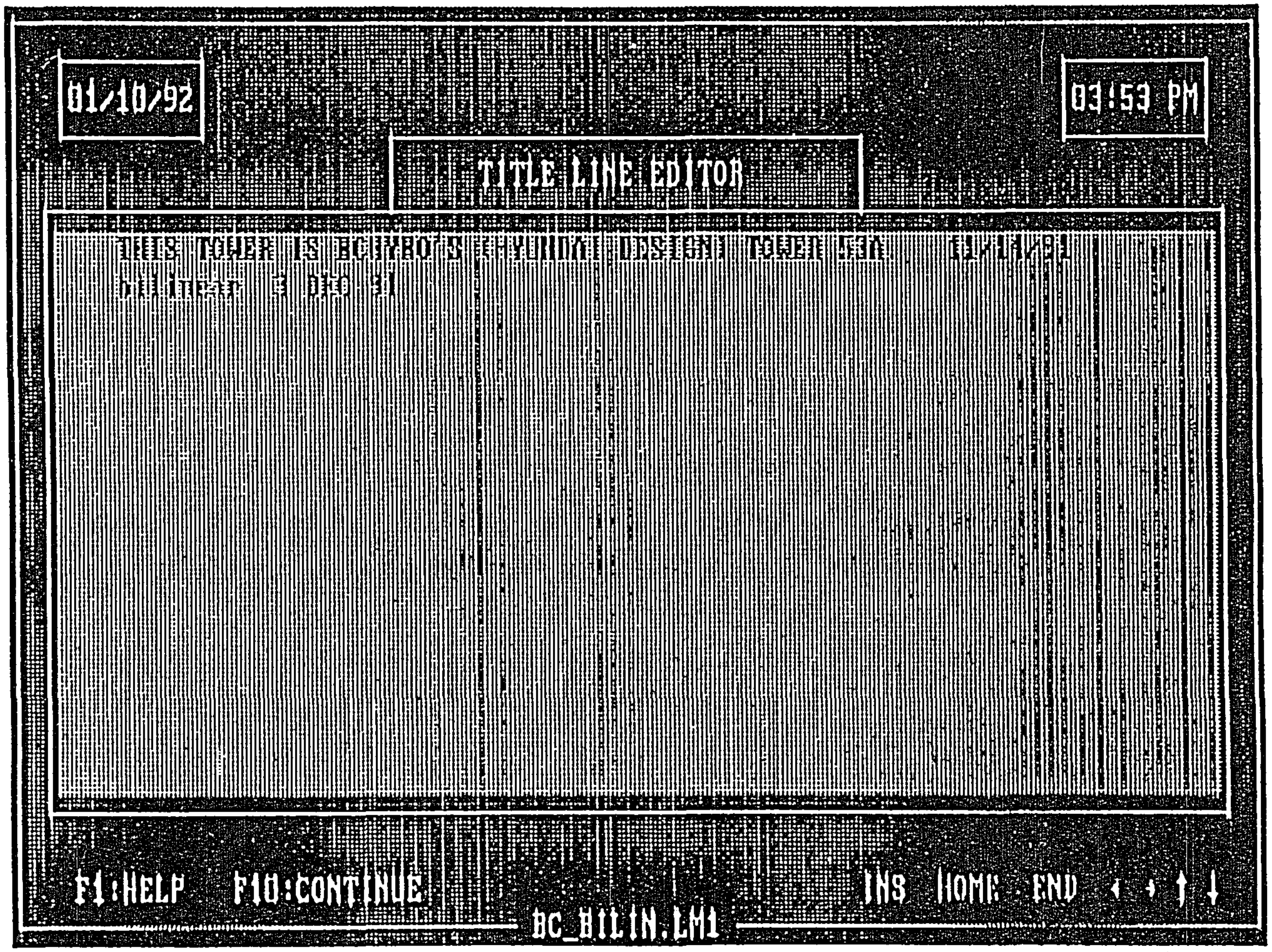


Figure 37. Limit States Analysis Module Title Line Editor.

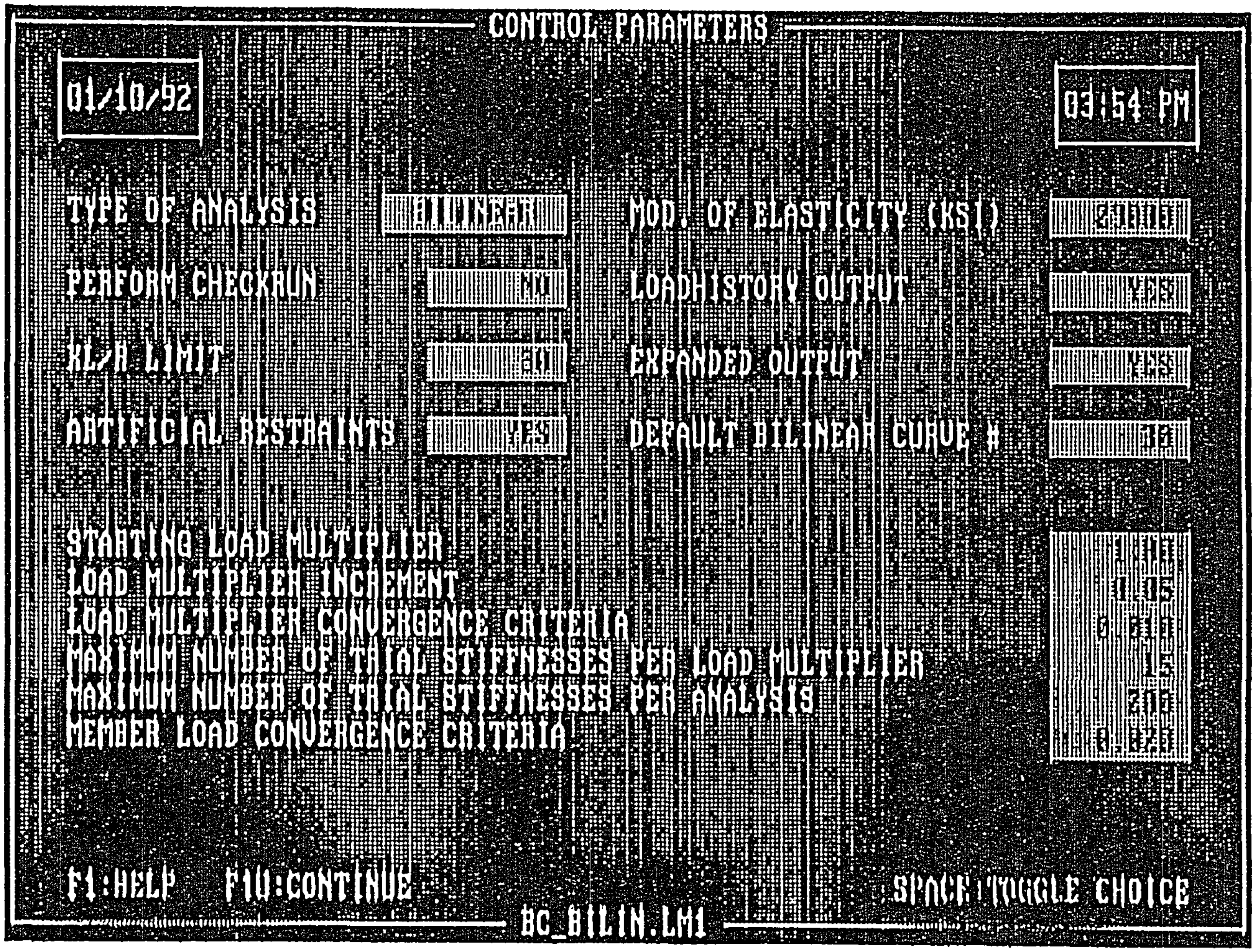


Figure 38. Limit States Analysis Module Control Parameters.

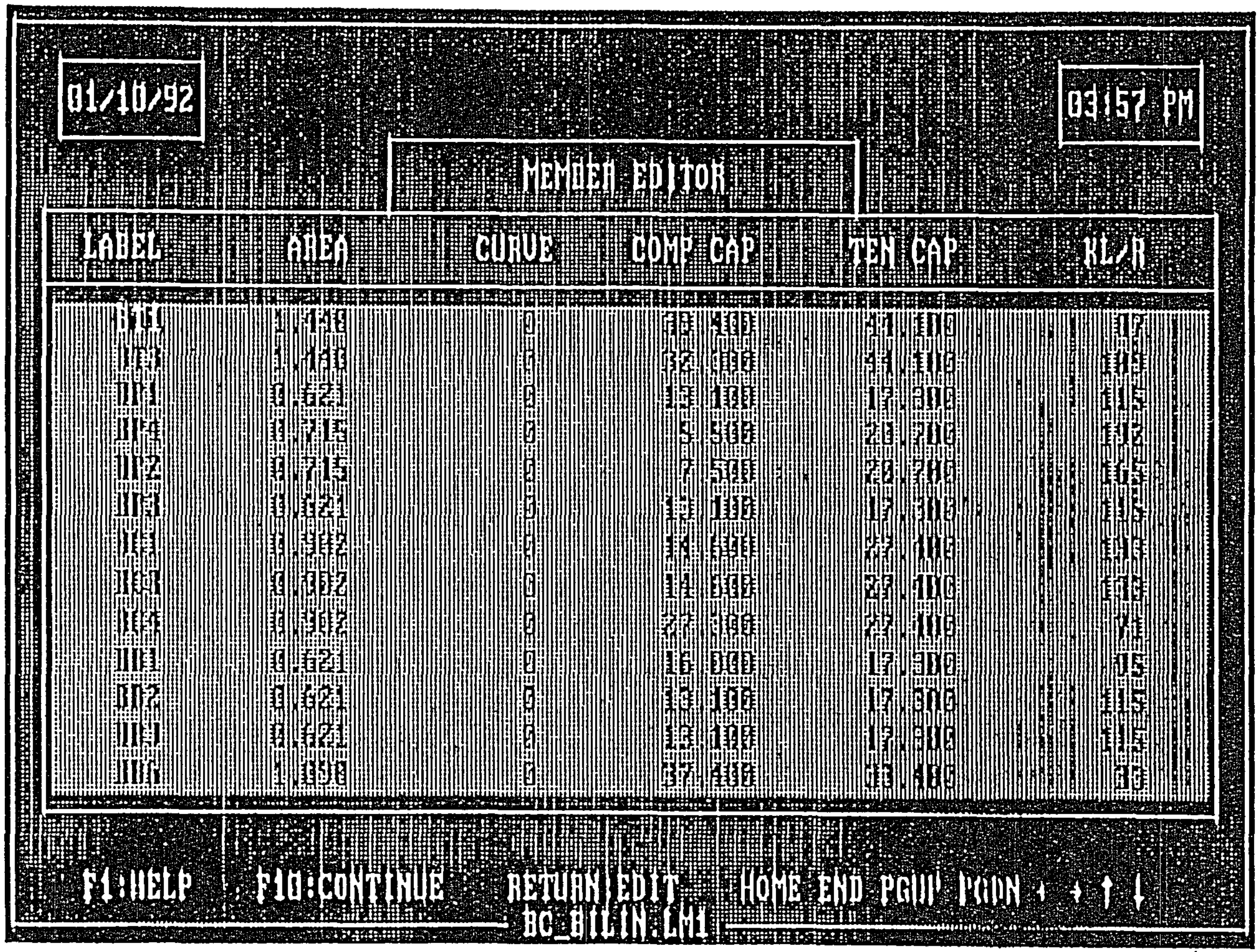


Figure 39. Limit States Analysis Module Member Editor.

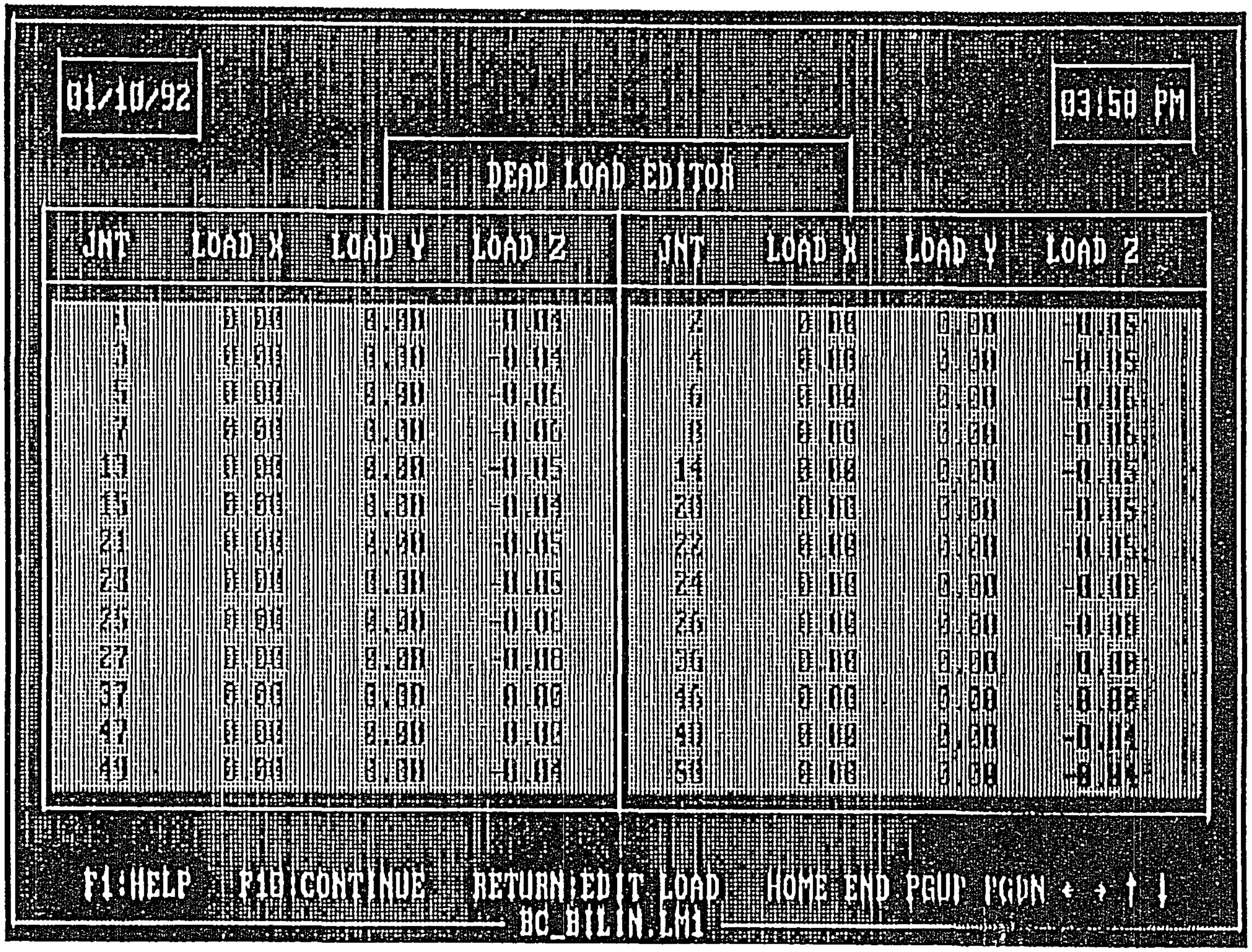


Figure 40. Limit States Analysis Module Dead Load Editor.

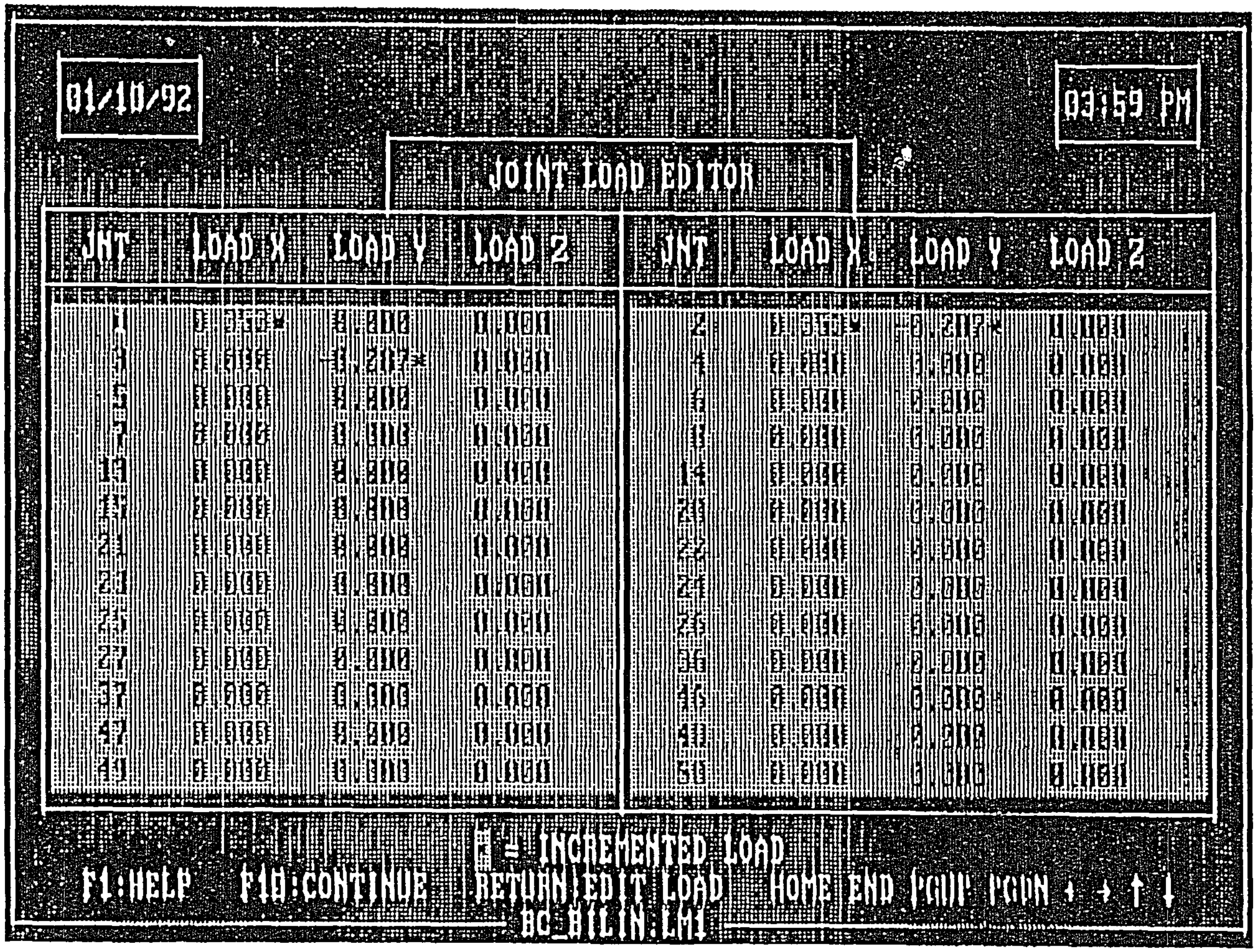


Figure 41. Limit States Analysis Module Joint Load Editor.

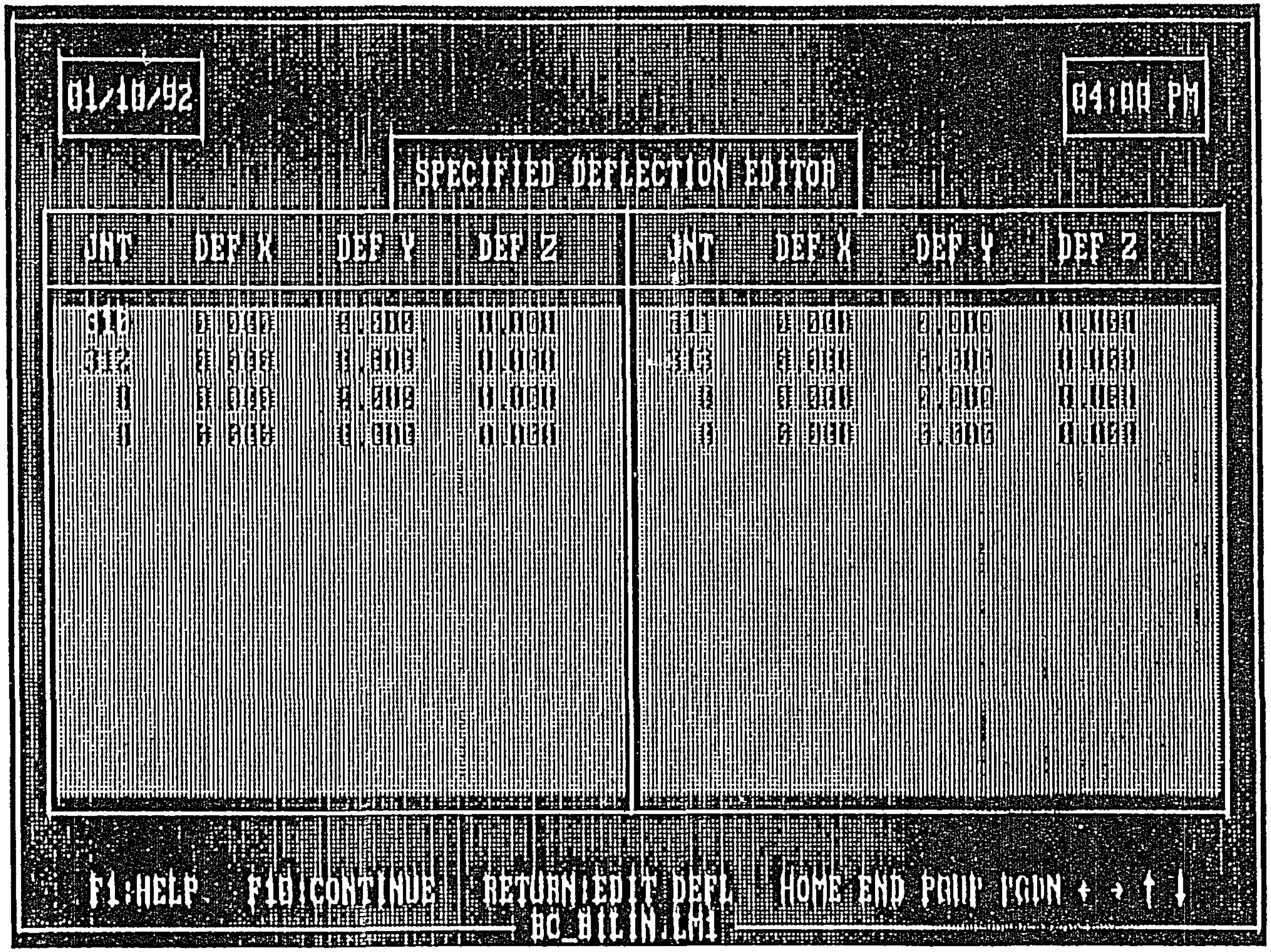


Figure 42. Limit States Analysis Module Specified Deflection Editor.

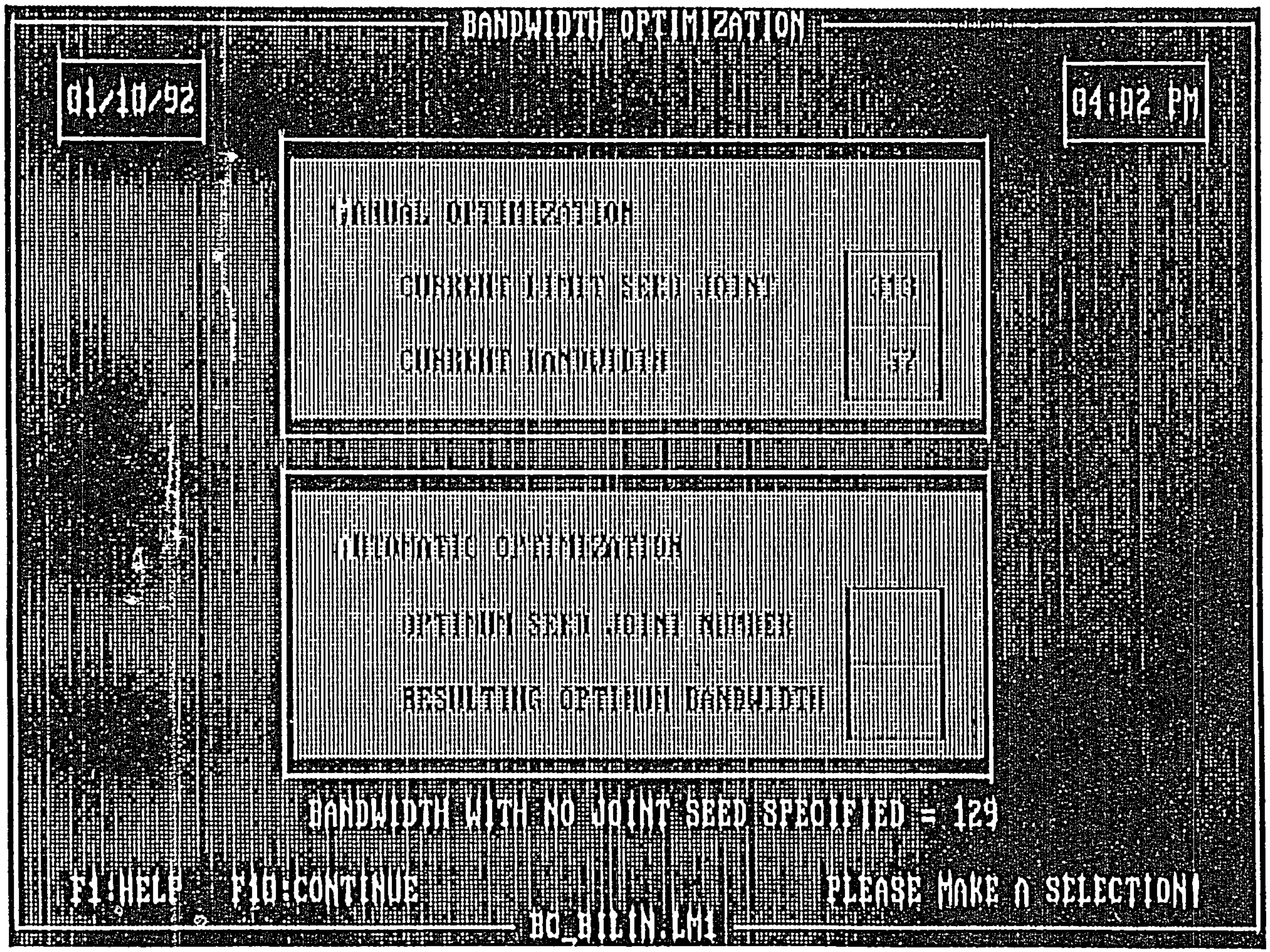


Figure 43. Limit States Analysis Module Bandwidth Optimization.

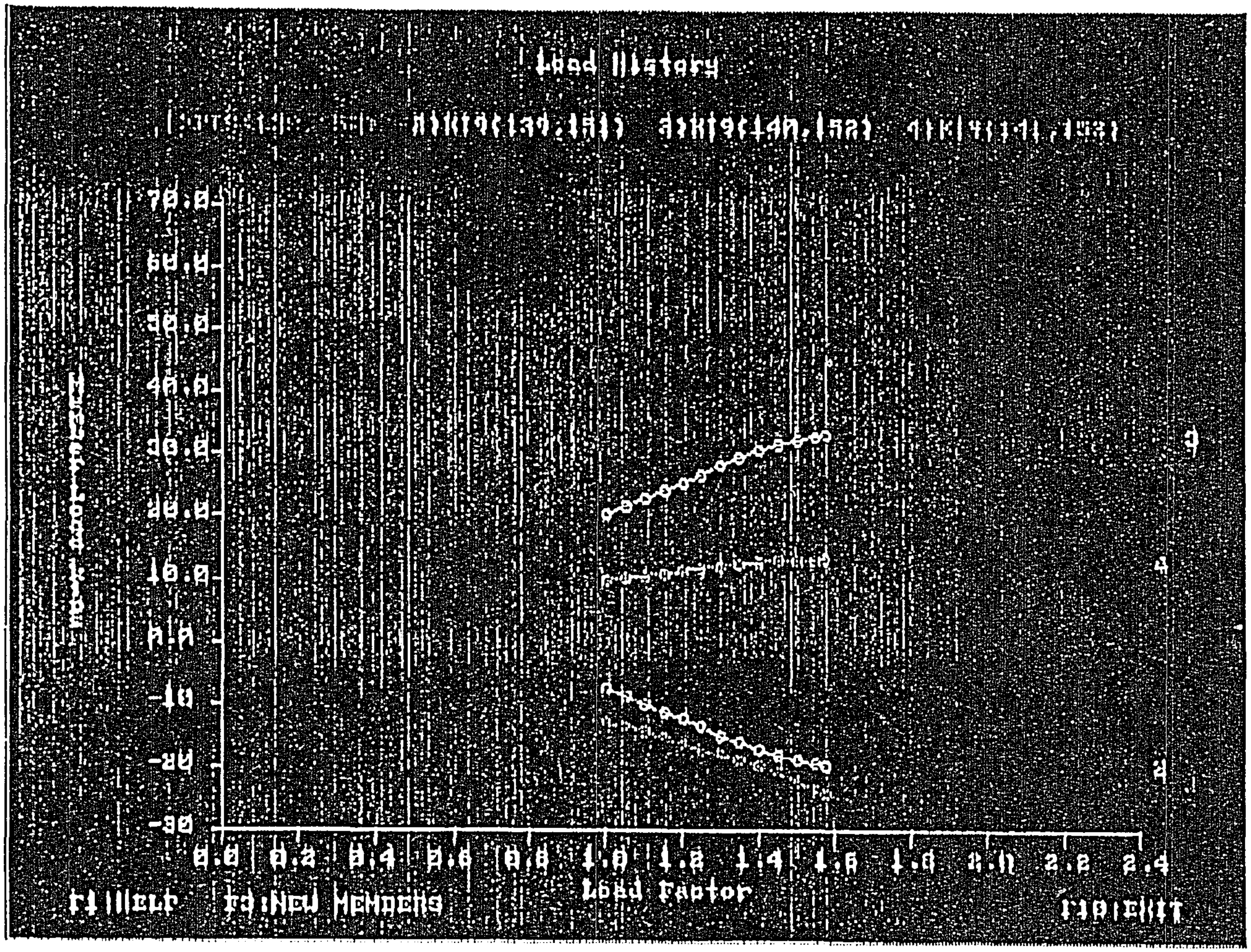


Figure 44. Limit States Analysis Module Load History.

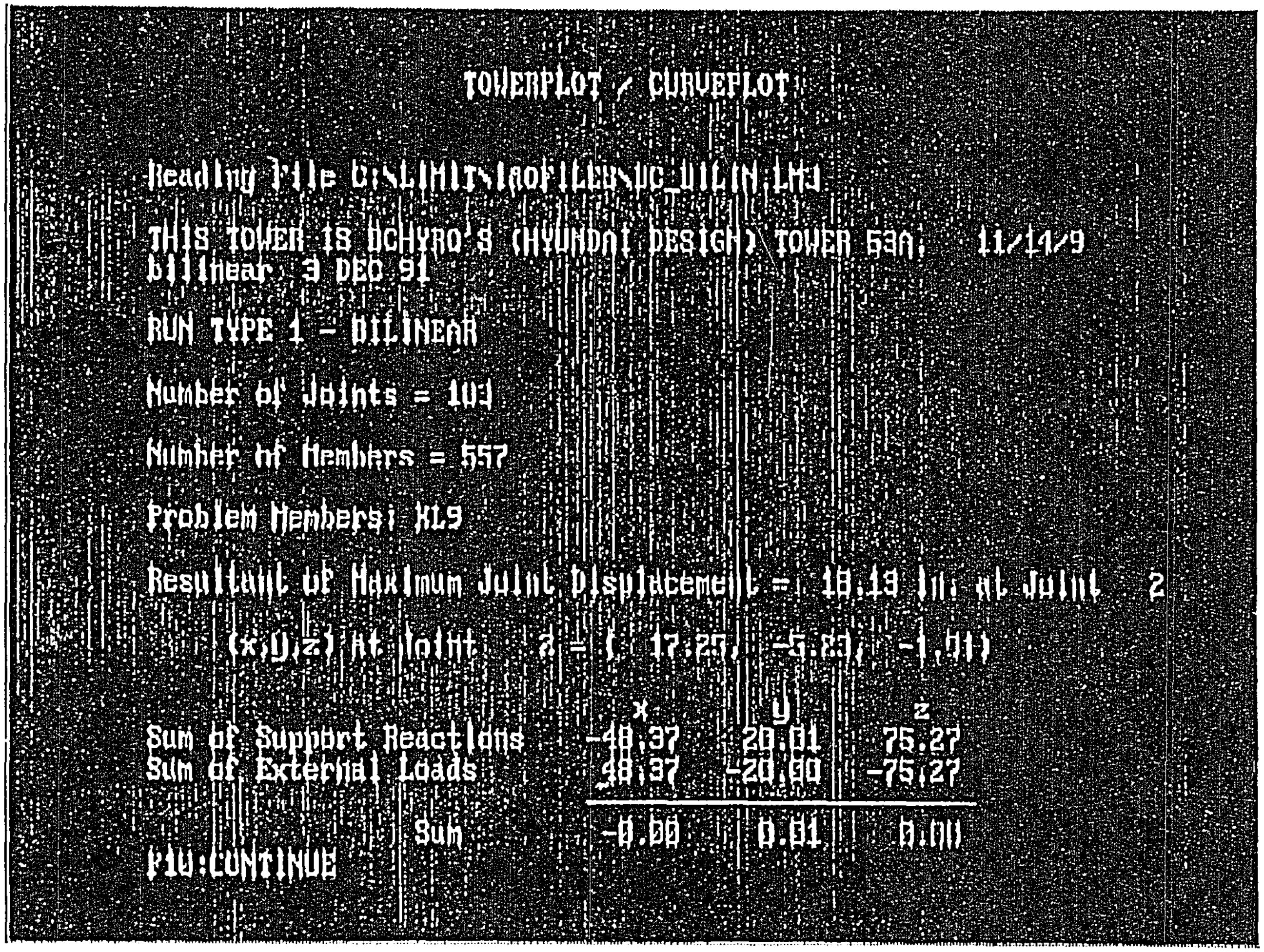
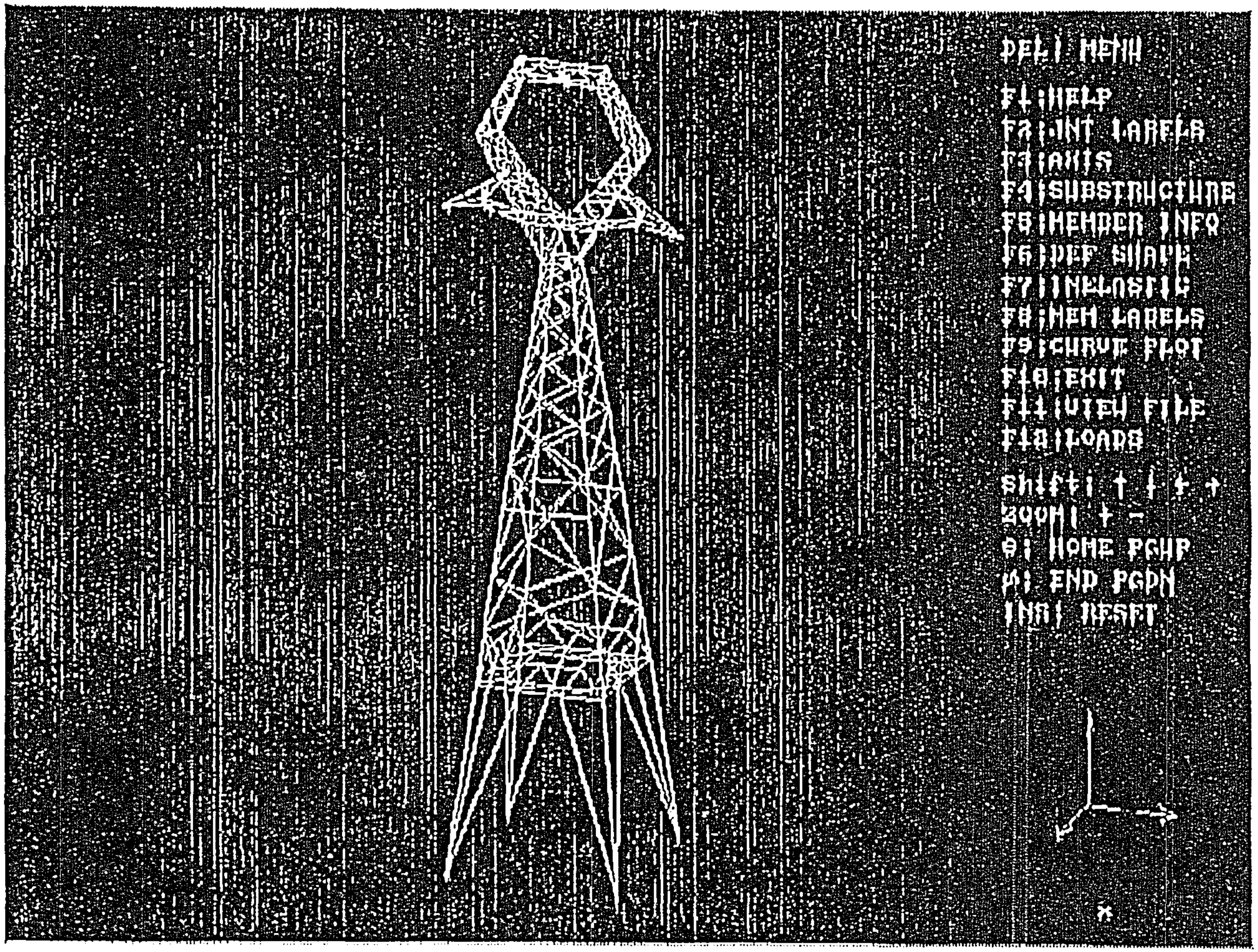


Figure 45. Limit States Analysis Module Towerplot\Curveplot Opening.



<u>Figure 46.</u> Limit States Analysis Module Towerplot.

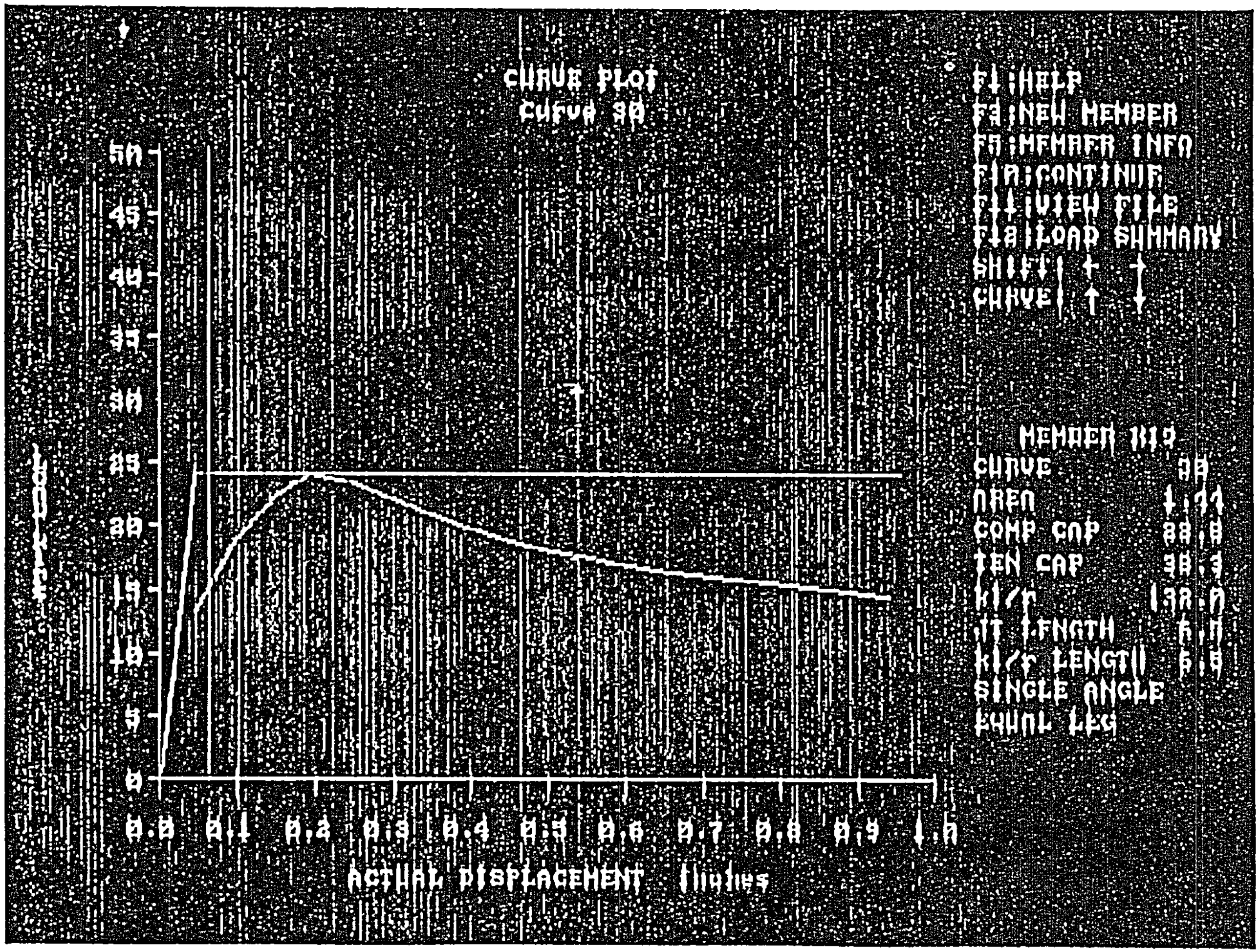


Figure 47. Limit States Analysis Module Curveplot.

APPENDIX D

FILE AND DIRECTORY STRUCTURE

APPENDIX D

FILE AND DIRECTORY STRUCTURE

Directory Structure

C:\LIMIT	HELP
	MISC
	I&OFILES
	NXP
	TEMP

Files contained in C:\LIMIT

START.EXE	TOWER.EXE	FILES.TMP
CONTROL.EXE	RUNBAT.TMP	DBOS_RES.COM
LIMIT4.EXE	C.BAT	DISCLAIM.EXE
MENU.BAT	NAME.EXE	LOADHIST.EXE
CONVERT.EXE	LIMIT4.OBJ	LIMIT4.FOR
KILL_DBO.COM	USER.EXE	STARTLIM.BAT
MAIN.OBJ	DBOS.LIB	LIMIT4.INF
HELVB.FON	MAINMENU.EXE	DBOS_SET.COM
LIMIT4.LIS	E.BAT	EXPERT.BAT
MODEL.EXE	COMP4.BAT	L.BAT
DBOS.EXE	RUN4.BAT	

Files contained in C:\LIMIT\HELP

MODFIG1.PCX	MODFIG2.PCX	MODFIG3.PCX
MODFIG4.PCX	MODFIG5.PCX	MODFIG6.PCX
MODFIG7.PCX	MODFIG8.PCX	MODFIG9.PCX
MODFIG10.PCX	MODFIG11.PCX	MODFIG12.PCX
MODFIG13.PCX	MODFIG14.PCX	MODFIG15.PCX
MODFIG16.PCX	MODFIG17.PCX	MODFIG18.PCX
MODFIG19.PCX	MODFIG20.PCX	MODFIG21.PCX
MODFIG22.PCX	MODFIG23.PCX	MODFIG24.PCX

MODFIG25.PCX	USFIG1.PCX	USFIG2.PCX
USFIG3A.PCX	USFIG3B.PCX	USFIG3C.PCX
USFIG3D.PCX	USFIG3E.PCX	USFIG4.PCX
USFIG5.PCX	USFIG6.PCX	BAND_OPT.HLP
CON_PARA.HLP	DL_EDIT.HLP	IN_DIR.HLP
IN_MENU.HLP	JL_EDIT.HLP	MAIN_MNU.HLP
MEM_INFO.HLP	OUT_DIR.HLP	SD_EDIT.HLP
TL_EDIT.HLP	VIEW_PRN.HLP	MODEL.TXT
USER.TXT	HHISGRAP.TXT	

Files contained in C:\LIMIT\MISC

PZP.COM	PZP.PZD	PZP.PZO
PZP.PZX	PZP1.PZD	PZP2.PZD
VPIC.CFG	VPIC.DOC	VPIC.EXE

Files contained in C:\LIMIT\NXP

EXPIN.KB	MESSAGE1.TXT	TEMPLTE1.TXT
EXIN.RTD	NXPFORMS.EXE	NXPFORMS.DOC
NXPFORMS.DAT	NEXPERT.DAT	NEXPERT.DOC
EXPOUT.KB	EXOUT.RTD	MESSAGE2.TXT
NXPINST.EXE	NXPMSVR.EXE	TEMPLTE2.TXT

Files contained in C:\LIMIT\I&OFILES

CURVEEXA.DAT	CURVENEW.DAT	CURVEOLD.DAT
CURVE30.DAT	CURVE.DAT	EXAMPLE1.LM1
EXAMPLE2.LM1	EXAMPLE3.LM1	EXAMPLE4.LM1

Files contained in C:\LIMIT\TEMP

INFILE.TMP	SUMMARY.TMP